

Comunicaciones industriales y WinCC

Luis Peciña Belmonte

marcombo FORMACIÓN

SIEMENS

Marcombo



¿Quieres adentrarte en la denominada **IV Revolución Industrial**?

La integración, la digitalización y la conectividad son los nuevos paradigmas de la nueva industria. Las comunicaciones industriales van a tener un papel principal; Internet y la nube son ya parte del presente.

Las redes industriales basadas en Ethernet, como Profinet, están experimentando un gran avance ya que son redes que se adecuan a los nuevos tiempos. Los SCADAS, el OPC, Internet de las Cosas (IoT), las redes AS-i, Profinet y Profibus, el Wifi industrial y la interactividad con las redes sociales, como Twitter, son parte de la nueva era de la digitalización y son aspectos que se tratan en este libro.

La gran experiencia como profesor del autor, de más de 30 años enseñando a jóvenes profesionales del Centro Salesianos de Zaragoza, hace de este texto un **manual eminentemente práctico**, donde se realizan muchas configuraciones y aplicaciones, con una descripción clara y sencilla. En el libro se recogen:

- Ejercicios de WinCC en TIA PORTAL.
- Actividades de Profibus, Profinet y AS-i en diferentes configuraciones con el PLC S7-1500 de Siemens.
- Lenguaje AWL para la implementación de cada ejercicio.
- Implementación de aplicaciones con otros dispositivos de otros fabricantes y los PLC's S7-300 y S7-1200 de Siemens.
- Scadas con el uso de WinCC y DSC de National Instruments (en el entorno de LabVIEW).
- Ejercicios novedosos con el Internet de las Cosas, utilizando el SIMATIC IoT 2040.
- Descripción de la conexión a Internet de sistemas de comunicación industrial y el envío de mensajes de texto a móviles (SMS) y e-mails desde distintos dispositivos.

Además, en la parte inferior de la primera página encontrará el código de acceso que le permitirá descargar de forma gratuita los contenidos adicionales del libro en **www.marcombo.info**.

Este manual va dirigido a los profesionales que, desconociendo este apasionante mundo, desean introducirse en las comunicaciones industriales. También se destina a aquellos iniciados que buscan adentrarse en aspectos como el acceso al Internet de las Cosas (SIMATIC IOT2000) en la industria.

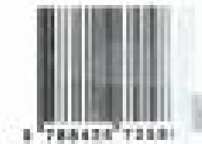
De igual modo, es adecuado para los alumnos que están cursando el Ciclo Formativo de Automatización y Robótica Industrial, para alumnos de Grado Universitario de Mecatrónica y, en general, para técnicos de cualquier especialidad interesados por temas tan actuales, y con tanto futuro, como los tratados en este libro.

No esperes más: forma parte del futuro inmediato. ¡Intégrate en la **IV Revolución Industrial**!





www.marcombo.com



COMUNICACIONES INDUSTRIALES Y WINCC

Luis Peciña Belmonte



Comunicaciones industriales y WinCC

Primera edición, 2018

Diseño de la cubierta: ENEDENÚ DISEÑO GRÁFICO
Maquetación: cuantofalta.es
Las imágenes contenidas en el libro son cortesía de Siemens, salvo donde se indique otra fuente.

ISBN: 978-84-267-2588-2
D.L.: B-3299-2018

Impreso en Ulzama
Printed in Spain

A mi hija

AGRADECIMIENTOS

Tengo que significar la gran aportación de conocimientos e ideas que he recibido por parte del soporte técnico de Siemens AG. Son muchas las personas que forman ese gran equipo, siempre dispuesto a solucionar las dudas que se presentan en la puesta al día de nuevos asuntos técnicos. Aun siendo un amplio equipo, me gustaría nombrar alguno de ellos, aquellos con los que he tenido un contacto más frecuente para la elaboración de este libro. Nombres como María Calavia, Luis Acosta u Oliver Neubauer son algunos de los que se merecen este agradecimiento por su grandísima paciencia hacia mi persona y por su buen hacer.

Deseo también agradecer la incansable ayuda que me ha ofrecido Laura Martín Castillo, del Departamento de Ingeniería de Digitalización de Siemens, en un tema de absoluta novedad como es el de la digitalización con SIMATIC IoT2000.

Igualmente quiero agradecer la entrega y apoyo, humano y técnico, recibido por el responsable de controladores Simatic y Proyecto SCE, Francisco Javier Cano Gatón. De la misma forma tengo que nombrar a Daniel García Martínez, siempre cerca de los centros de formación apoyando nuestra labor.

Mención importante merecen mis alumnos Álvaro Martínez Veraméndiz y Javier Blanco Espeleta por su buen hacer y su aportación a este libro.

PRÓLOGO

El **sector industrial** está desempeñando un papel cada vez más importante a nivel mundial. Se considera un **motor de la innovación, crecimiento y estabilidad social.** Sin embargo, la competencia es cada vez mayor. Los clientes demandan **productos de alta calidad y personalizados, con un tiempo de producción menor**.

Solo aquellos que logren conseguir productos customizados y que reduzcan el tiempo de producción, con la máxima eficiencia en sus plantas, serán capaces de ser competitivos. La **solución** estriba en la unión del mundo real y virtual, es decir, la **industria 4.0** y la digitalización.

Clave de la Industria 4.0 es la interacción/comunicación industrial entre todos los componentes del proceso productivo, desde el nivel de campo hasta los niveles IT.

Nos congratula comprobar cómo Luis Peciña, profesor de Formación Profesional de Salesianos Colegio Nuestra Señora del Pilar de Zaragoza, ha sido capaz en poco más de 300 páginas de ofrecer una visión detallada de las redes estándares de comunicación industrial: Profibus, Ethernet, WLan, Radiofrecuencia (SINAUT), así como de las de reciente implementación Profinet e IoT (Internet of Things).

El libro, al incorporar en su desarrollo los controladores avanzados **SIMATIC S7-1500,** base de la Industria 4.0, y **la Ingeniería TIA** Portal, que constituye la puerta de acceso a la Factoría Digital, tendrá vigencia durante muchos años.

Quiero felicitar y agradecer a Luis Peciña su esfuerzo en la edición de este libro, libro que será todo un referente en el mundo de la automática y una excelente herramienta de aprendizaje tanto para los actuales como futuros profesionales.

Deseo agradecer a la editorial Marcombo por su continuada apuesta en la divulgación de las enseñanzas técnicas en su vertiente más práctica.

Francisco Javier Cano Gatón
Responsable de Controladores SIMATIC y
Proyecto SCE – Siemens Automation Cooperates with Education
Division Digital Factory
SIEMENS, S.A.

Índice general

INTRODUCCIÓN

Al hablar de un tema como el de las comunicaciones industriales, siempre existe la tendencia de extenderse en aspectos muy teóricos y dejar la parte práctica relegada a un segundo plano. En este libro se tratará la parte teórica de forma muy breve, insistiendo en lo más importante, y se va a desarrollar extensamente la práctica. Se utilizarán los sistemas PLC de la serie 1500 y la versión V14 de TIA PORTAL.

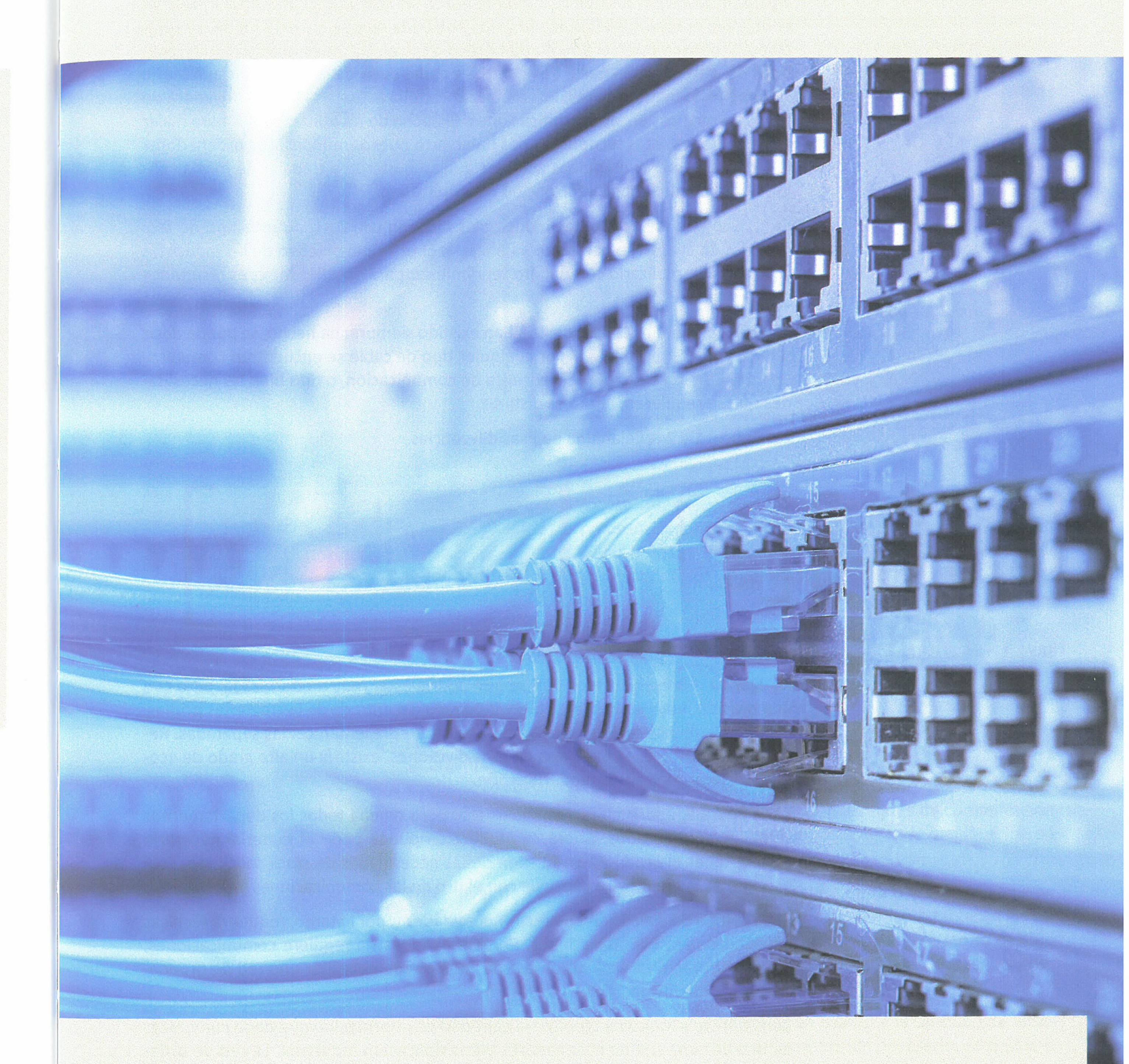
En el ambiente industrial de finales de esta segunda década del siglo XXI se observan una inmensidad de cambios, aunque muchos de ellos se aplican menos de lo que se oyen. Poco a poco se crean nuevas aplicaciones donde la digitalización desempeñará un papel prioritario. No es la primera vez que esta adquiere protagonismo en el mundo técnico. Ya en las últimas décadas del siglo XX se hablaba de digitalización. En ese momento, se modernizaron las transmisiones de las telecomunicaciones y, por supuesto, también los ambientes industriales con máquinas, como la introducción del PC en la industria y la expansión de los autómatas programables.

Hoy se vuelve a hablar de la digitalización y aparecen conceptos como el Internet de las cosas y el uso de la nube para la industria, entre otros. Todo esto aparece englobado dentro de lo que se ha determinado en denominar la «IV Revolución Industrial».

En este libro se hablará de todo ello, pero, sobre todo, se van a realizar ejercicios prácticos donde la interconexión de los sistemas automáticos constituirá el principal objetivo.

Se abarcarán las comunicaciones más actuales como Profinet, AS-i, redes wifi y Profibus. Esta última en clara decadencia por la expansión de las redes Profinet. También una parte importante se dedica a los sistemas HMI (Interfaz Humano Máquina). En este caso, se utiliza el *software* WinCC de Siemens.

Unidad 1 Principios básicos de comunicaciones



En esta unidad veremos:

1.1 Por qué las comunicaciones en la industria

Cuando los sistemas industriales se hicieron más complejos, también el cableado, entre ellos, requería de una inmensa cantidad de cables. Ese fue uno de los principales motivos por los que se implementaron las comunicaciones en la industria; el objetivo era reducir el cableado y no cabe duda de que, de este modo, se solucionó. Si pensamos en el ámbito doméstico o social, las redes sociales como Facebook o Twitter forman parte de nosotros y nadie duda de su importancia, aunque siempre existen detractores. Internet es la estrella actual y la gran motivadora de todos los avances en las comunicaciones. El hecho de no utilizar cables para hacer efectivas todas estas tecnologías nos ha abierto nuevos mundos. Los ordenadores fueron las primeras máquinas actuales que se pusieron a trabajar de forma conjunta intercambiándose información o compartiéndola.

Era lógico que los ambientes industriales hicieran uso de todo este tipo de tecnologías en cuanto se demostró su eficacia. El principal objetivo de las industrias es producir y realizarlo con un alto rendimiento para obtener mejores resultados económicos.

Las comunicaciones en la industria han existido siempre; de hecho, se puede decir que dos dispositivos unidos por cualquier tipo de cable se encuentran comunicados. Lo que sucede es que el concepto de comunicación, o más bien las necesidades que se requieren, ahora es distinto.

Las figuras 1 y 2 reflejan una de esas diferencias.



Figura 1

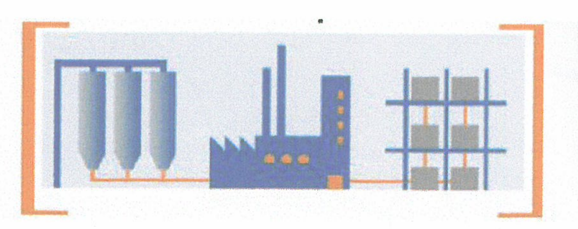


Figura 2

La figura 1 representa una empresa que podría estar muy modernizada, pero con una división muy estanca, muy individualizada. La entrada de las materias primas, el proceso de fabricación y el almacenamiento representan celdas separadas. Resulta evidente que, en cada una de esas divisiones, existiría un alto grado de «comunicación», incluso de modernización.

En la figura 2, aparece la misma empresa, pero en ella todos los departamentos se hallan comunicados. Ahora, la producción puede pedir el material necesario para obtener el producto en función de las necesidades de *stock* solicitado. Este ejemplo es lo que hoy se consigue con el uso de las comunicaciones industriales.

Cada campo de aplicación posee sus necesidades y, por lo tanto, requerirá técnicas adecuadas para sus aplicaciones. Por eso, la industria requiere comunicaciones, pero con características diferentes a las que puede necesitar una red de ordenadores en una oficina.

Considérese a modo de ejemplo la cantidad de información que debe comunicar de una vez una red de ordenadores y una aplicación industrial. La red de ordenadores de una oficina o de un despacho de CAD ha de intercambiar grandes cantidades de datos; una aplicación industrial, no. Imaginemos que se quiere transmitir el estado de un sensor *on/off* (final de carrera, por ejemplo), pues con un bit ya resulta suficiente. Claro está que no siempre será un bit; debe tomarse como un ejemplo extremo. Sin embargo, en una comunicación industrial puede ser importante conocer el tiempo en que se va a producir la comunicación efectiva y que no varíe, dentro de un margen conocido. Pongamos por caso que ese sensor establece que un ascensor está en una planta determinada y debe pararse; si lo hace tarde o cada vez en un tiempo diferente, imaginamos lo que

Recuerda . . .

En las comunicaciones industriales, los conceptos y características son distintos a otro tipo de comunicación como, por ejemplo, las redes de ordenadores. También las necesidades resultan diferentes, en función de la aplicación de esas comunicaciones.

pasará: la puerta puede no abrirse o hacerlo cada vez en una posición distinta con el consiguiente riesgo.

Con esto, lo que se quiere destacar es que habrá necesidades distintas para cada aplicación donde intervengan las comunicaciones.

Se pueden ver los niveles de aplicación en la llamada «pirámide de la automatización». En ella se establecen las distintas necesidades dependiendo de las aplicaciones para cada sección. Hay que tener en cuenta que para que, la comunicación sea completa y óptima, debe ser vertical entre los diferentes niveles, y horizontal, entre los diferentes dispositivos de cada nivel. En la figura 3 se muestra la pirámide de automatización.

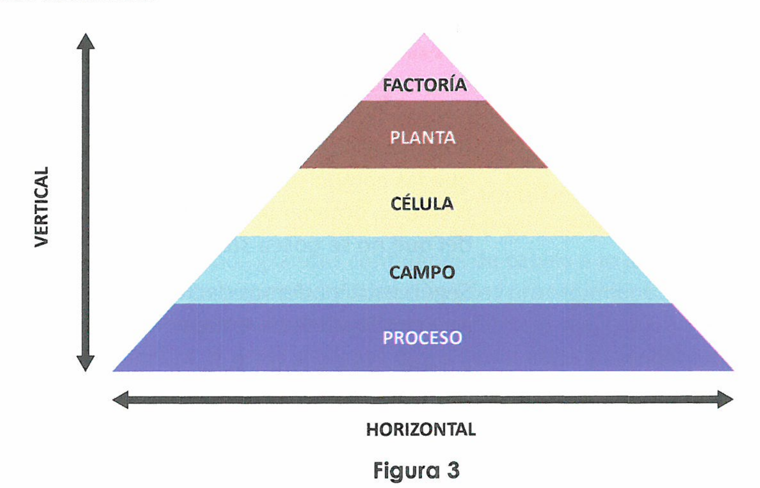


Figura 3

El nivel de Campo y Proceso constituye el nivel más bajo de la automatización, ya que en él residen los procesos de producción y, por lo tanto, es el terreno apropiado para el control de las máquinas que trabajan ahí. Por ello, aquí se hallarán los autómatas, los buses de campo, los sensores y actuadores más directos para el control de la producción.

En el nivel de Célula se coordina y organiza todo el control que se realiza en los niveles anteriores. Aquí se dispone de redes de comunicación y de los PLC u ordenadores encargados de la coordinación. Se controla igualmente la secuenciación de la fabricación.

En el nivel de Planta se prepara el trabajo para las etapas inferiores. Se diseña, se organiza y prepara la programación que, después, se llevará a cabo en los PLC y robots. Los departamentos de CAD/CAM, de programación de robots y PLC y la oficina técnica se sitúan en este nivel. Se utilizan grandes redes de comunicación y bases de datos. Los SCADAS poseen un papel importante en la planificación de las tareas que se realizan.

En el nivel de Factoría se realiza la gestión completa de la producción. Aquí se produce la comunicación con los proveedores y los clientes y se establecen las órdenes para el diseño y la producción. La gerencia y las oficinas tendrán su destino principal en este nivel. Se necesitan, igualmente, grandes ordenadores, servidores, bases de datos y redes de comunicación.

En cada uno de estos niveles existen diferencias en cuanto a la cantidad de información que se debe transmitir y el tiempo de transmisión necesario. Este último hay que considerarlo como el tiempo de respuesta, es decir, lo que le cuesta responder a un dispositivo desde que se le da la orden.

Se ha de observar que el tiempo de transmisión cercano a la máquina (final de carrera, por ejemplo) tiene que ser casi instantáneo, de unos pocos milisegundos. En los niveles superiores de la pirámide, esos tiempos se muestran superiores. Respecto al volumen de datos resulta a la inversa. En los niveles superiores, se requiere utilizar grandes bases de datos y, por lo tanto, se precisa de un gran volumen de información; entre otros, suministrado por los niveles inferiores.

Para que esta pirámide sea efectiva, la comunicación en este tipo de sistemas debe ser total. Eso significa que la información ha de fluir en todos los sentidos, desde los niveles más bajos a los superiores (comunicación vertical), pero también entre todos los participantes de un mismo nivel (comunicación horizontal).

El señor gerente del nivel de factoría no se interesa por la información concreta que proporcione un sensor concreto; si está o no activado no le incumbirá, pero sí resultan de gran importancia la información que se reúna con dicho sensor. Esa información, una vez tratada, puede ofrecer detalles, por ejemplo, de las paradas de la producción que se han producido en un periodo de tiempo, o del número de botellas que se han embotellado en dicha franja.

1.2 Ventajas de las comunicaciones en la industria

Con todo lo dicho anteriormente, se pueden deducir algunas de las ventajas de lo que las comunicaciones consiguen en la industria. El objetivo de toda producción reside en obtener un producto de calidad a un buen precio, que sea competitivo y del que no se posea *stock* almacenado.

Según esto, podemos decir que, con las comunicaciones en la industria, se consiguen las siguientes ventajas:

- Reducción de los *stocks*.
- Posibilidad de descentralizar la producción en varias plantas, al trabajar conjuntamente.
- Aumento de la eficacia de los diferentes procesos que intervienen en la producción.
- Control automático de las necesidades de material en función de los pedidos.
- Comunicación automática con los pedidos de los clientes.
- Reducción de costes de mantenimiento, instalación y puesta a punto.
- Trazabilidad de la producción desde el principio: desde el pedido del producto hasta su entrega.
- Mayor control de los procesos intermedios, así como sus alarmas.
- Mantenimientos preventivos basados en hechos «históricos».

El uso de las comunicaciones en la industria puede aumentar la producción con unos menores costes y un incremento de la calidad. De esta forma, también se puede ser más competitivo.

En la figura 4 se resume parte de lo descrito en este apartado.

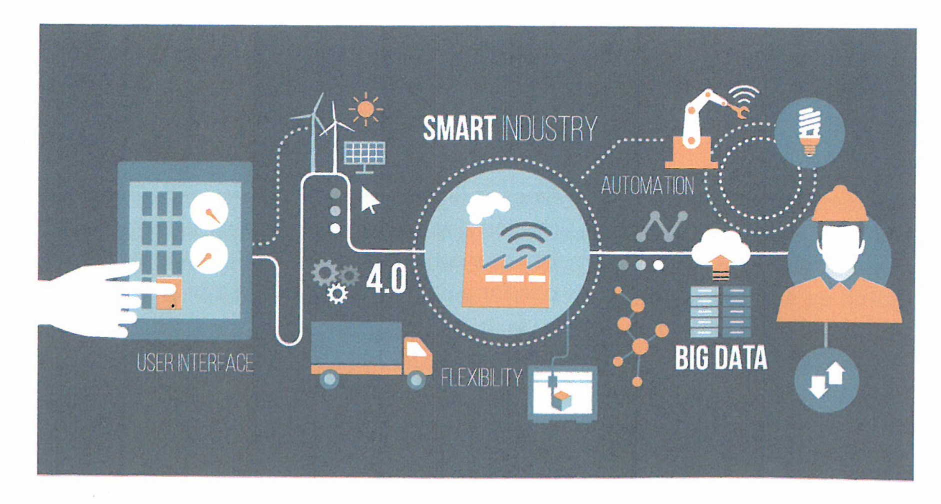


Figura 4

1.3 Sistemas distribuidos, centralizados y abiertos

Hoy día cuando hablamos de sistemas de comunicación industrial, nos referimos a *sistemas distribuidos* y también a *sistemas abiertos*. Se trata de dos conceptos diferentes, pero muy relacionados. Esto es más cierto cuanto mayor se muestra la industria.

Si hablamos de pequeñas empresas, las comunicaciones se refieren casi exclusivamente a una máquina. Estos sistemas solo comunican un PLC, o poco más, con la máquina. Muchas veces ni siquiera se utiliza una comunicación industrial; simplemente se cablean las E/S a la máquina directamente mediante un gran número de cables. Estos sistemas se llaman *centralizados*. En dicho caso, las entradas y salidas se encuentran junto al propio PLC. Los PLC se concentran en armarios desde los cuales salen gran cantidad de cables que van de las unidades de E/S a los sensores y actuadores, siempre alejados de estos armarios.

En los sistemas de comunicación actual, se reparten las tareas entre diferentes PLC; todos estos autómatas se encuentran comunicados mediante un único **cable de comunicación** (evitamos grandes cantidades de cables a lo largo de la fábrica). Este reparto de tareas hace que la información y los procedimientos se *distribuyan* a lo largo de la fábrica e incluso entre diferentes fábricas deslocalizadas por países y continentes distintos. A estos sistemas que utilizan estos procedimientos de comunicación se los denomina ***SISTEMAS DISTRIBUIDOS;*** resulta lo habitual hoy día.

En las figuras 5 y 6 se observan sistemas distribuidos.

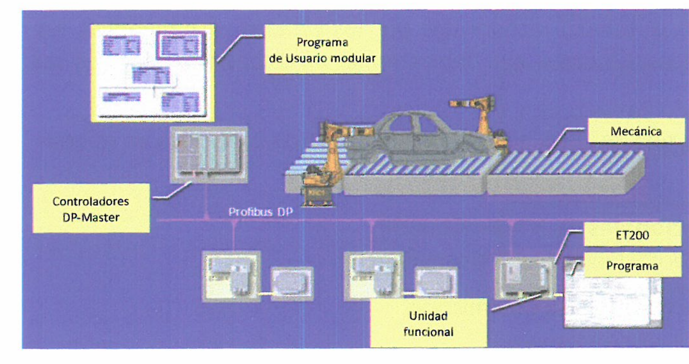


Figura 5

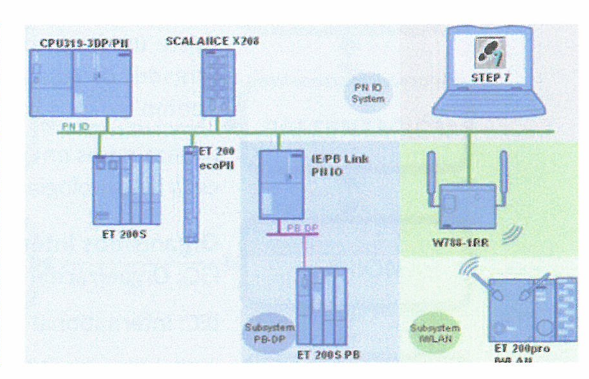


Figura 6

Los *sistemas distribuidos* tienen como características propias, además de lo dicho anteriormente para las comunicaciones industriales, las siguientes:

- Integración y convivencia de diferentes sistemas de comunicación, con estándares diferentes y, por lo tanto, no compatibles directamente.
- Utilización de grandes sistemas complejos: cuanto más grandes, más clara la necesidad de estos sistemas.
- Reparto de las responsabilidades, lo que también hace más sencillo acotar los problemas.
- Utilización de una red de tipo industrial.
- Aumento de la flexibilidad en la fabricación, ya que podemos parar parte del proceso sin necesidad de detener todo el sistema. Lo mismo se puede aplicar a ampliaciones o reparaciones.
- Incremento a medida de las necesidades de producción o económicas.
- Integración en ***sistemas abiertos***; es decir, que pueden utilizarse dispositivos (PLC o dispositivos de E/S) de fabricantes diferentes, ya que los estándares de comunicación no pertenecen a ningún fabricante. Cualquiera que cumpla con la normativa del estándar puede vender un producto.

Recuerda . . .

La estandarización es un concepto muy deseado en la industria. En las comunicaciones industriales, se disponen de diversos estándares muy populares y aceptados. Estos se hallan coordinados por diferentes organismos nacionales e internacionales.

1.4 Normalización y estandarización en las comunicaciones

En las comunicaciones, como en cualquier otro ámbito, hay una normativa que cumplir para que los fabricantes se acojan a ella y puedan poner sus productos al servicio de la empresa.

Desgraciadamente, en comunicaciones industriales no existe una sola norma, tampoco un solo estándar. Curiosamente, hay muchos estándares. Lo bueno es que, en la actualidad, estos sistemas son abiertos; es decir, no pertenecen a un solo fabricante. Dichos estándares son agrupaciones de empresas que crean el estándar, la norma. Posteriormente, es el mercado el que determina, con su uso o no, si el sistema resulta aceptado. Claro está que este grupo de empresas que inician el estándar serán compañías de gran nivel internacional. Posteriormente, cualquier fabricante, acogiéndose a la norma, podrá desarrollar productos y sacarlos al mercado una vez realizados los ensayos pertinentes. Los sistemas abiertos o no propietarios inciden directamente en la competencia y, por ello, también en la lucha de precios y calidades. De esta forma, el mercado dispone de variedad de precio y características.

Lo contrario a los sistemas abiertos son los sistemas propietarios, aquellos que son propios de un fabricante y que se deben adquirir obligatoriamente a un único fabricante. Todos conocemos ejemplos de este tipo de actividad y sabemos lo que conlleva.

1.4.1 Organismos de normalización

Existe un grupo de organismos, reconocidos por los Estados o continentes, encargados de establecer las normas que debe cumplir un dispositivo para que sea óptimo.

Realizaremos un resumen de los diferentes organismos más conocidos en referencia a la tecnología.

Organismos Internacionales de Normalización

ISO: Organización Internacional para la Normalización.

IEC: International Electrotechnical Commission.

IEEE: Institute of Electrical and Electronics Engineers.

ITU: Unión Internacional de Telecomunicaciones (engloba CCITT y CCIR).

Organismos Europeos de Normalización

CENELEC: Comité Européen de Normalisation Électrotechnique (Comité Europeo de Normalización Electrotécnica).

CEN: Comité Europeo de Normalización.

Organismos de Estados Unidos

ANSI: American National Standards Institute.

EIA: Electronics Industries Alliance.

Organismos españoles

AENOR: Agencia Española de Normalización.

1.4.2 Normas relacionadas con las comunicaciones

EIA RS232: norma de la comunicación en serie RS232.

EIA RS422: norma de la comunicación en serie RS422.

EIA RS485: norma de la comunicación en serie RS485.

IEEE 802: Redes de Área Local (LAN).

CENELEC EN50170: buses de campo industriales de propósito general.

CENELEC EN 61131-5: comunicaciones en los autómatas programables.

IEC 61158: buses de campo industriales.

1.5 Modelo OSI para sistemas abiertos

Como ya se ha dicho, los sistemas de comunicación industrial hoy día son abiertos, es decir, que no tienen un propietario, por lo que, cualquiera puede fabricar acogiéndose a una norma concreta.

Para definir la norma de los sistemas abiertos, se halla el modelo OSI, que es el acrónimo de Open System Interconection.

Este modelo dispone de siete capas o niveles y, en cada uno de ellos, se define cómo debe comportarse la comunicación. Pueden verse los niveles en la figura 7. Se aprecian los siete niveles para establecer la comunicación entre dos interlocutores: A y B.

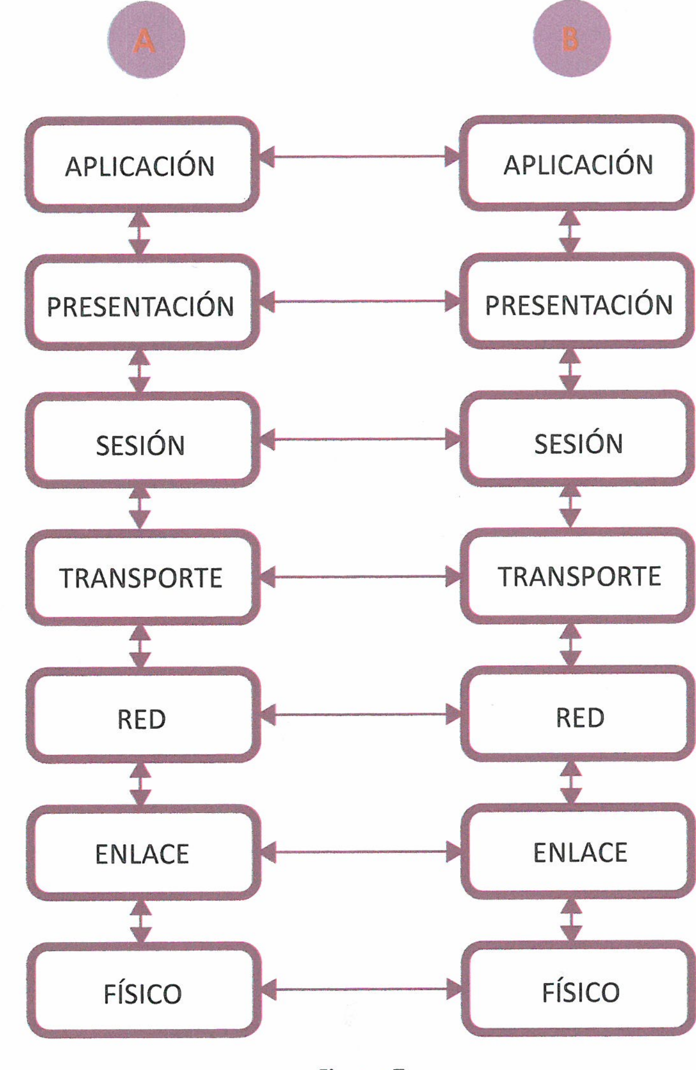


Figura 7

Dentro de cada nivel se establecerá el protocolo, esto es, el conjunto de reglas que se deben cumplir para un tipo de comunicación determinada. No resulta obligatorio que se cumplan los siete niveles. De hecho, en comunicaciones industriales lo normal es que se cumpla tan solo con tres niveles; siempre dependerá de la com-

Recuerda . . .

Todos los estándares de comunicación industrial deben cumplir con el modelo OSI de sistemas abiertos, aunque sea parcialmente.

plejidad de la comunicación. A estos niveles el usuario no accede, pues se hace solo mediante programas informáticos, que son los que ejecutan las reglas de cada nivel.

Capa física

De todas las capas, la física es la más fácil de comprender ya que reúne todos los aspectos materiales de la conexión, como el tipo de cableo, los niveles eléctricos de transmisión. Define la forma de conectarse a los otros dispositivos. Además de lo referente al medio físico, también determina la forma en la que se transmite la información.

Las funciones más importantes son:

- Definir el medio o medios físicos por los que va a viajar la comunicación: cable de pares trenzados, de cobre normal, no trenzado, coaxial, aéreo, fibra óptica.
- Determinar las características materiales (componentes y conectores mecánicos) y eléctricas (niveles de tensión) que se van a usar en el envío de los datos por los medios físicos.
- Transmitir el flujo de bits a través del medio.
- Manejar las señales eléctricas del medio de transmisión o polos en un enchufe.
- Garantizar la conexión (aunque no la fiabilidad de dicha conexión).

Capa de enlace de datos

Una vez establecido el medio físico, el siguiente paso es direccionar la información; por ello, esta capa se ocupa del direccionamiento físico, del acceso al medio, de la detección de errores, de la distribución ordenada de tramas y del control del flujo.

En esta capa se segmenta la información en tramas para ser trasladadas de una forma «empaquetada» (paquetes de información, es decir, grupos de información).

También se verifica la información y se corrigen errores y, por ello, es importante mantener una excelente adecuación al medio físico con el medio de red que redirecciona las conexiones mediante un *router*.

El dispositivo que usa la capa de enlace es el *switch*, que se encarga de recibir los datos del *router* y enviar cada uno de estos a sus respectivos destinatarios, esto es, los diferentes dispositivos conectados a la red.

Capa de red

Se encarga de identificar el camino existente entre una o más redes. Las unidades de información se denominan «paquetes».

El objetivo de la capa de red es hacer que los datos lleguen desde el origen al destino, aun cuando ambos no se encuentren conectados directamente. Los dispositivos que facilitan tal tarea se denominan *routers*. Los *firewalls* (cortafuegos) actúan sobre esta capa principalmente, para descartar direcciones de máquinas.

En este nivel se realiza el direccionamiento lógico y la determinación de la ruta de los datos hasta su receptor final.

Capa de transporte

Esta capa es la encargada de efectuar el transporte de los datos (que se encuentran dentro del paquete) de la máquina de origen a la de destino y lo independiza del tipo de red física que esté utilizando.

Sus protocolos son TCP y UDP: el primero orientado a conexión y el otro sin conexión. En el TCP (orientado a la conexión) el envío de información solo se produce si el destinatario se halla conectado. En el UDP (orientado a la desconexión) se producirá siempre el envío, aunque el destino no esté conectado. Un ejemplo de

UDP sería el correo electrónico. Si no se encuentra conectado, el mensaje siempre llega al buzón y se recoge la información cuando el usuario se conecta. Como TCP se puede considerar la descarga de un documento desde Internet; si no se halla conectado, el envío se interrumpe.

Capa de sesión

No es una capa imprescindible y, de hecho, muchas veces se ignora. Esta capa se encarga de mantener y controlar el enlace establecido entre dos computadores que están transmitiendo datos de cualquier índole. Por lo tanto, el servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, esta se pueda efectuar para las operaciones definidas de principio a fin y reanudarla en caso de interrupción. Esta función resulta intrínseca a la capa de transporte.

Capa de presentación

Esta capa se encarga de poner en orden toda la información que va a llegar. Los dos equipos no tienen por qué ser idénticos y por ello, aunque distintos equipos presenten diferentes representaciones internas de caracteres, los datos deben llegar de manera reconocible.

Esta capa es la primera en trabajar más el contenido de la comunicación que el cómo se establece. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras cuentan con diferentes formas de manejarlas.

Esta capa también permite cifrar los datos y comprimirlos. Por lo tanto, podría decirse que esta capa actúa como un traductor.

Capa de aplicación

Las diferentes aplicaciones que se disponen en un PC deben permitir acceder a los servicios de las capas anteriores. Esta capa define los protocolos de las aplicaciones para intercambiar datos con el correo electrónico (SMTP), los gestores de bases de datos y los servidores de ficheros (FTP), etcétera.

El usuario no accede a la capa de aplicación directamente, lo hace a través de los programas que sí conectan con la capa de aplicación, haciendo que el usuario se aleje de la complejidad de los protocolos.

Se puede resumir de un modo sencillo las características de cada capa de este modo:

Capa de aplicación: programas de usuario.

Capa de presentación: estandarización de la presentación a las aplicaciones.

Capa de sesión: gestión de las conexiones entre aplicaciones.

Capa de transporte: conexión física y proporciona el envío de datos.

Capa de red: gestión de las conexiones hacia las capas superiores.

Capa de enlace de datos: detección y corrección de errores.

Capa física: red material.

El modelo OSI en las comunicaciones industriales se condensa a tres capas. Son comunicaciones más sencillas; por eso se reducen a las capas física, de enlace y de aplicación.

1.6 Transmisión en serie y en paralelo

Hoy día, la comunicación en paralelo solo se utiliza entre dispositivos internos en los PC o PLC. La comunicación en paralelo se caracteriza por enviar muchos datos de una vez. En la comunicación en serie, los datos se transportan de uno en uno, mediante una señal de sincronismo que marca la pauta de envío. Resulta evidente que la comunicación en paralelo es más rápida para volúmenes grandes de datos.

Con la aparición de la comunicación en serie USB, la transmisión en paralelo casi ha dejado de existir para interconectar dispositivos externos.

Todas las comunicaciones industriales son del tipo serie, con lo que evitamos transportar grandes cantidades de cables por las fábricas, de modo que se ahorra materiales, en mantenimientos y en puestas en marcha de las instalaciones.

1.6.1 Sincronización en serie: síncrona y asíncrona

La capa de enlace de datos tiene, entre otras funciones, la de realizar la sincronización de los datos, es decir, poner de acuerdo al emisor y receptor para enviar y recibir la información. Se debe saber cuándo se empieza y cuándo se termina de mandar los datos.

Existen dos procedimientos: de forma asíncrona y síncrona. En la **sincronización asíncrona,** la forma de organizar el envío de los datos se incluye en el mismo dato; unos bits son de sincronización y otros, el propio dato.

En este tipo de envío, no se conoce con precisión cuándo se recibirá el mensaje. Cada carácter que se desee transmitir incluye un bit de información denominado «de arranque», y uno o dos bits de parada.

- El bit de arranque posee la función de sincronización de reloj del transmisor y del receptor. El reloj marca la cadencia con la que se envía o reciben los datos y cada uno, emisor y receptor, utiliza el suyo propio.
- El bit o los bits de parada se usan para separar un carácter del siguiente.

Después de la transmisión de los bits de información, se suele agregar un bit de paridad (par o impar). Dicho bit sirve para comprobar que los datos se transfieran sin errores. El receptor revisa la paridad de cada unidad de entrada de datos.

Cuando el emisor tiene el nivel lógico 1 (parada), el emisor informa al receptor de que va a llegar un carácter; para ello antepone un bit de arranque *(Start)* con el valor lógico 0. Una vez que el bit *Start* llega al receptor, este disparará un reloj interno y se quedará esperando los sucesivos bits que contendrán la información del carácter transmitido por el emisor. En la figura 8 se ve la secuencia.

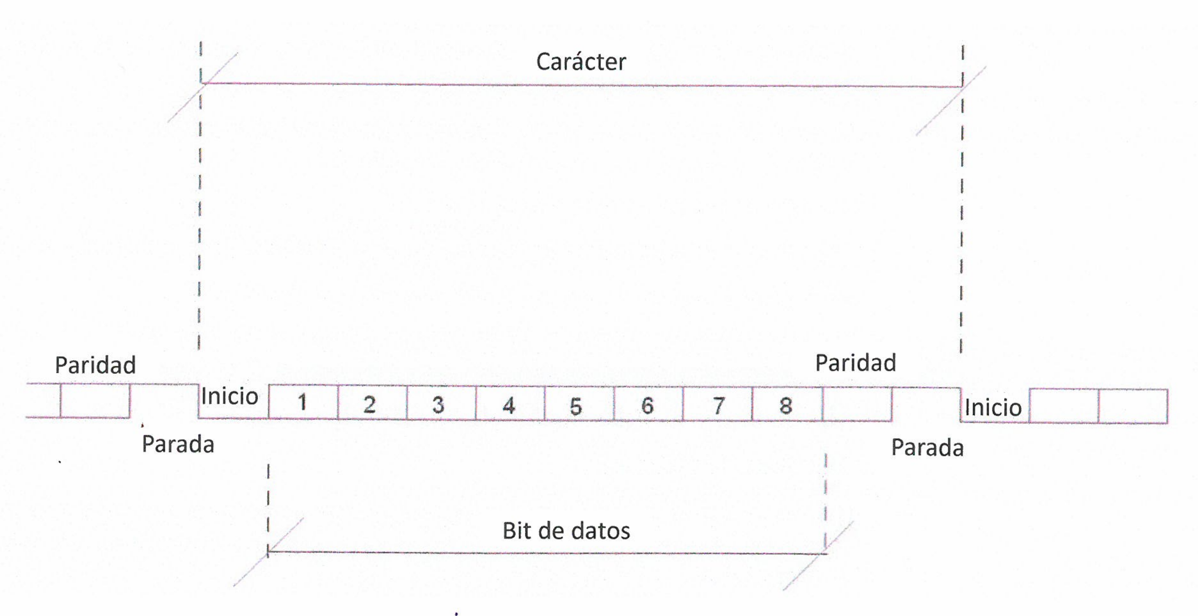


Figura 8

Como el emisor y el receptor cuentan con sincronizaciones independientes, se pueden producir errores.

En la **sincronización síncrona,** el reloj dispone de una señal independiente y se transmite a la vez que los datos, o bien se envía una trama especial de sincronización del reloj interno.

Recuerda . . .

La topología de red más utilizada en la industria es la de bus. Cada dispositivo que participa en una red se denomina «nodo».

Aquí se transmite un bloque de bits sin utilizar códigos de comienzo o parada. Para prevenir la desincronización entre el emisor y el receptor, se deben sincronizar sus relojes de alguna manera. Una posibilidad estriba en proporcionar la señal de reloj a través de una línea independiente.

Uno de los extremos (el receptor o el transmisor) enviará regularmente un pulso de corta duración que servirá para sincronizar sus relojes. El otro extremo utilizará esta señal a modo de reloj. Si la distancia es grande, también se producirán retrasos, por lo que esta forma de transmitir solo se utiliza en distancias cortas.

La otra alternativa consiste en incluir la información relativa a la sincronización en la propia señal de datos.

Full duplex y half-duplex

El intercambio de datos sobre una línea de transmisión se puede clasificar como *full-duplex* y *half-duplex.* En la transmisión *half-duplex,* se consigue comunicación en los dos sentidos (envío y recepción), pero no simultáneamente. Un ejemplo podría ser las comunicaciones entre los antiguos radioaficionados (los *walkie-talkies*).

En la transmisión *full-duplex,* las dos estaciones pueden simultáneamente enviar y recibir datos. El ejemplo para este caso lo constituiría la conversación telefónica.

En la transmisión *full-duplex* normalmente se exigen dos caminos separados (por ejemplo, dos pares trenzados), mientras que la transmisión *half-duplex* necesita habitualmente uno.

1.7 Topología de redes

Cuando se habla de topología, nos referimos a la forma en que está diseñada la red, bien físicamente (referido al *hardware*) o bien lógicamente (referido al *software*).

En una topología de red, cada dispositivo participante y enlazado en la red (PLC, PC) se denomina «nodo». La topología es la forma geométrica de la relación entre todos los enlaces y los nodos.

Las principales topologías de red son la malla, la estrella, el **árbol**, el bus y el anillo. En la actualidad las más utilizadas son las de bus y anillo.

Topología punto a punto

En este tipo solo dos dispositivos se encuentran en comunicación. Se trata de la más simple de todas las posibles (figura 9).



Figura 9

Topología en malla

En una topología en malla, cada nodo presenta un enlace punto a punto. El tráfico de información se produce únicamente entre los dos dispositivos que se encuentran en comunicación. A este tipo de comunicación se le denomina «enlaces dedicados» (figura 10).

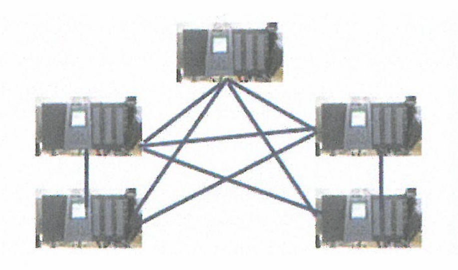


Figura 10

Una ventaja se halla en que, al realizarse la comunicación exclusivamente entre dos nodos, solo estos poseen acceso a la información y no el resto. Otra ventaja reside en que el fallo de un enlace no afecta al resto de los participantes en la red.

Topología en estrella

La topología en estrella es similar a la de en malla, ya que cada dispositivo solamente cuenta con un enlace punto a punto, pero, en este caso, pasa por un controlador central, llamado «concentrador». Este sirve de intermediario entre los enlaces punto a punto. El nodo que quiere enviar lo hace al concentrador y este lo manda al receptor de destino (figura 11).

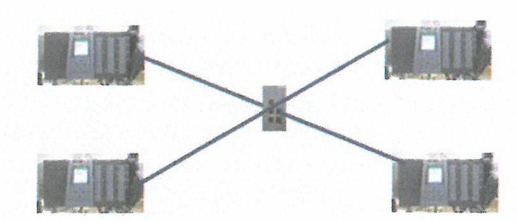


Figura 11

En una red de estrella, cada dispositivo necesita solamente un enlace y un puerto de entrada/salida para conectarse a cualquier número de dispositivos.

Topología en árbol

La topología en árbol es una variante de la de en estrella. Aquí hay más de un concentrador y cada uno se comunica con los demás. En este caso, cada rama/concentrador presenta nodos colgando de él (figura 12).

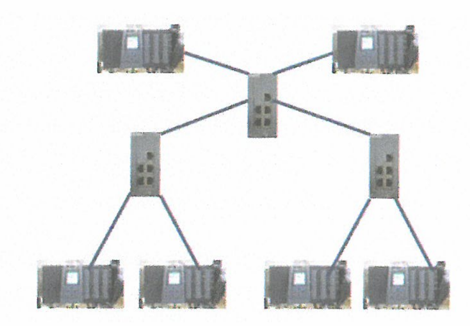


Figura 12

El controlador central del **árbol** es un concentrador repetidor, es decir, que regenera la información para poder acceder a una mayor distancia y que no se pierdan datos.

Topología en bus

Una topología en bus es multipunto. Un cable largo actúa como una red troncal que conecta todos los dispositivos en la red (figura 13).

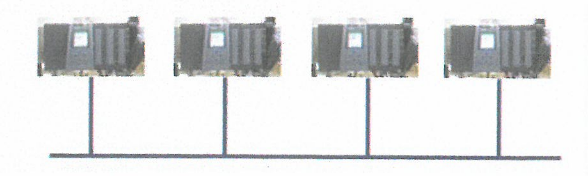


Figura 13

Los nodos se conectan al bus mediante cables de conexión (latiguillos) y conectores apropiados.

Entre las ventajas de la topología de bus, se incluye la sencillez de instalación y el ahorro de cable de red.

En este caso debe realizarse un equilibrado de la red, de forma que los extremos dispongan de una impedancia que iguale a la de los demás nodos.

Topología en anillo

En una topología en anillo, cada dispositivo tiene una línea de conexión dedicada y punto a punto solamente con los dos dispositivos que se hallan a sus lados. La señal pasa a lo largo del anillo en una dirección, o de dispositivo a dispositivo, hasta que alcanza su destino. Cada dispositivo del anillo incorpora un repetidor (figura 14).

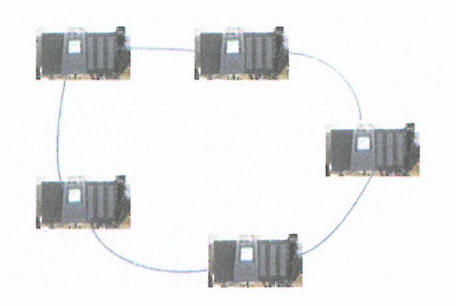


Figura 14

Un anillo es relativamente fácil de instalar y reconfigurar. Cada dispositivo se enlaza solamente con sus vecinos inmediatos (bien físicos o lógicos). Para añadir o quitar dispositivos, tan solo se deben mover dos conexiones.

1.8 Métodos para acceder al medio

En aquellas comunicaciones donde existan muchos participantes, y el medio sea común para todos, se ha de organizar de alguna forma el procedimiento para acceder. La capa de enlace del sistema OSI se encarga de definir estos métodos. En las comunicaciones punto a punto, no existen tales problemas al solo existir un emisor y un receptor. En cualquier comunicación, sea del tipo que sea, solo un nodo emite y otro recibe; los demás se hallan a la espera.

Existen dos métodos para acceder al medio:

- Centralizado.
- Contienda.

1.8.1 Sistemas de acceso centralizados

Es de esperar que, en un tipo de comunicación centralizado, solo haya un dispositivo central de control. Este se encarga de organizar quién puede comunicar.

Existen dos sistemas centralizados:

- Sondeo/selección.
- Paso de testigo.

Los sistemas que utilizan el **sondeo** son aquellos en los que un equipo de la red (Maestro/Master) «pregunta» a cada participante (Esclavo/Slave) y, si cuenta con información, la transmite. Si el **sistema** es **de selección**, el maestro selecciona con qué esclavo quiere comunicarse. Este sistema sirve para distribuir los esclavos (unidades pasivas de E/S); es, por tanto, un sistema monomaestro. Un maestro será siempre un dispositivo inteligente, que dispone de capacidad de procesamiento, como un PLC. Un esclavo normalmente será una estación pasiva (terminal tonto), una unidad de E/S sin capacidad de procesar/decidir.

Este tipo de acceso se emplea en Profibus DP.

En los sistemas de **paso de testigo,** no existe solo un maestro; se puede decir que todos son maestros. De cada maestro puede colgar una red de esclavos. En este caso, de forma cíclica se pasa una palabra binaria (Trama) con información de control. Esta palabra puede ser de información o de testigo. Cada equipo, cuando

Recuerda . . .

En las comunicaciones industriales los estándares más extendidos solo utilizan la capa física, la de enlace y la de aplicación.

recibe la palabra como testigo, puede enviar información si lo desea; si no fuese así, pasaría a la siguiente estación. Si desea enviarla, quitará la codificación de testigo y la pondrá de información. También aparecerá la dirección del dispositivo con el que quiere comunicar. Todos los equipos recibirán esa palabra, pero solo la albergará aquel que posea la dirección a la que va dirigida dicha información. Cuando el equipo de destino recibe la trama, asoma un ACK (un OK de recibido). A continuación, la trama volverá al emisor tras comprobar el ACK y, por lo tanto, que la información ha llegado al destino. Ahora este equipo será el encargado de volver a poner la palabra en formato de testigo para comenzar un nuevo ciclo.

Pudiera suceder que, cuando una estación recibe la trama testigo, no tuviera información que comunicar; entonces, pasará la trama testigo a la siguiente estación.

Este sistema es el utilizado en Profibus FMS.

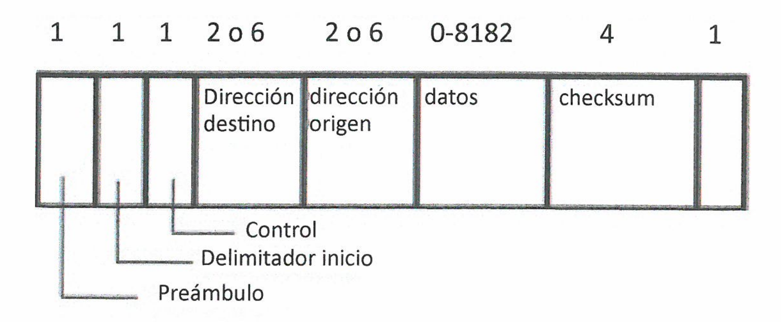


Figura 15

En la figura 15 puede verse la trama del sistema *token bus*. El control indica si la trama es de testigo (bit 09) o de datos (bit 1).

En estos métodos, el proceso de paso de información se muestra lento, puesto que debe sondearse o recorrer todas las estaciones.

1.8.2 Sistemas de acceso con contienda

En los sistemas de acceso con contienda, ya no existe un equipo encargado de la organización de la comunicación. Cada estación controla su propia comunicación. Todos los participantes intentarán enviar sus datos a la red.

El método más empleado es el CSMA (Carrier Sense Multiple Access) o Acceso Múltiple Sensible a la Portadora. Se trata de «escuchar» antes de «hablar». Escucha si la red está libre y, si es así, habla; es decir, envía la información. En el caso de que esté ocupada, espera. Puede suceder que se desee enviar dos estaciones simultáneamente, con lo que se producirá una colisión de datos. Para que las consecuencias de este hecho no sean importantes en la comunicación, se dispone de dos sistemas: uno es el CA (Collision Avoidance; en castellano, Prevención de Colisión) y el otro CD (Collision Detection, Detección de Colisión). La diferencia entre estos dos enfoques se reduce al envío —o no— de una señal de agradecimiento por parte del nodo receptor:

- ✓ **Collision Avoidance (CA)**: es un proceso en tres fases en las que el emisor primero escucha si se encuentra libre la red. Si lo está, transmite y espera a recibir la conformidad de llegada por parte del receptor. De esta forma existe seguridad de que llega la información. El problema reside en que se carga la red de datos por el hecho de transmitir la recepción de confirmación de llegada. La red Ethernet utiliza este método.
- ✓ **Collision Detection (CD):** con este método se envía sin escuchar si está libre la red; además, no se recibe confirmación de llegada. El método para saber si ha llegado a su destino la información es no «oír» colisión. Si no existe colisión, es que el dato ha llegado al destino.

En los dos métodos, cuando se produce una colisión, se vuelve a repetir el envío transcurrido un tiempo, hasta que los datos se manden correctamente.

Tal vez parezca que estos sistemas no funcionan bien, pero es como realmente se producen las comunicaciones en las redes más habituales. En la práctica no se producen casi colisiones y, si las hay, el sistema reacciona y envía rápidamente.

1.9 Sistemas determinísticos y no determinísticos

En muchas aplicaciones industriales resulta muy importante que, cuando se utilice una comunicación, se sepa con cierta precisión el tiempo en que se va a producir. Cuando se habla de «el tiempo en que se va a producir», nos referimos no tanto a un tiempo sino a un margen de tiempo; es decir, un tiempo máximo en que se asegure que la comunicación se va a realizar. Los sistemas en los que se realiza la comunicación en un tiempo conocido y no aleatorio son los llamados sistemas *DETERMINÍSTICOS*. Los estudiados anteriormente de *token passing* y los de sondeo pertenecen a este tipo, ya que se conoce el tiempo que costará sondear o pasar el testigo por todas las estaciones posibles. Si hubiera menos estaciones en la red, el tiempo sería menor.

En el caso de los sistemas de CSMA, lo típico son las colisiones y no se posee certeza de cuándo se va a poder efectuar la comunicación. Estos sistemas son los denominados *NO DETERMINÍSTICOS*.

Se debe destacar que los sistemas no determinísticos pueden mostrarse más lentos que los determinísticos, como acabamos de ver en las diferentes formas de acceder al medio. En los sistemas determinísticos, no se trata de que sean muy rápidos; más bien, lo que se desea es conocer el tiempo, o el intervalo de tiempo, en que se va a producir la transferencia de información.

El determinismo y el *tiempo real* se hallan muy relacionados en las comunicaciones industriales. No puede existir tiempo real si no hay determinismo. Tiempo real no es que los sistemas sean rápidos; más bien, el tiempo real requiere conocimiento del tiempo, es decir, determinismo.

Ejemplos donde el determinismo adquiere suma importancia puede ser la sincronización mecánica multieje a través de una red, las redes de seguridad o los *safety buses*; más concretamente, por ejemplo, en la industria del papel, donde grandes rodillos deben sincronizarse con precisión.

1.10 Interconexión de redes

Resulta vital en comunicaciones industriales la posibilidad de integrar en una misma instalación distintas tecnologías. En el caso de los dispositivos, se decía que los sistemas de comunicación industrial eran abiertos. En esa línea es muy importante poder enlazar y, por lo tanto, mantener en comunicación redes distintas, incluso con protocolos diferentes. Con ellos conseguiremos escalar (ampliar) nuestra instalación industrial según vayan variando las necesidades de la empresa.

Para ello se dispone de una serie de dispositivos que enlazan redes según las necesidades de interconexión:

REPETIDOR – *HUB* – PUENTE O *BRIDGE* – CONMUTADOR (*SWITCH*) – *ROUTER* – PASARELA O *GATEWAY*

Repetidores (*repeaters*)
Un repetidor es un dispositivo que repite todas las señales de un segmento de la red a otro en el ámbito eléctrico. Se utiliza para resolver dos problemas típicos en una red: el exceso de ordenadores conectados o la pérdida de la señal por la excesiva longitud del cable. No tiene en cuenta la dirección de la estación y envía

lo que recibe, incluidos los errores que puedan producirse en la comunicación. Un repetidor, por tanto, actúa solo en el nivel físico o capa 1 del modelo OSI.

Únicamente tienen una entrada y una salida (figura 16).

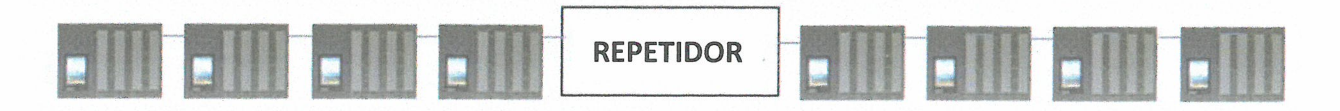


Figura 16

Hubs
Hoy día no se utilizan *hubs*. Un *hub* sirve para realizar una conexión en estrella y ampliar una red.

En Hub todos los nodos se conectan a él y, por lo tanto, tendrá que tener tantas «bocas» como nodos. Cuando un dispositivo quiere transmitir, se envía al *hub* y él lo envía al resto de participantes. En este caso, todos reciben la información.

Un *hub* actúa solo en el nivel físico o capa 1 del modelo OSI (figura 17).

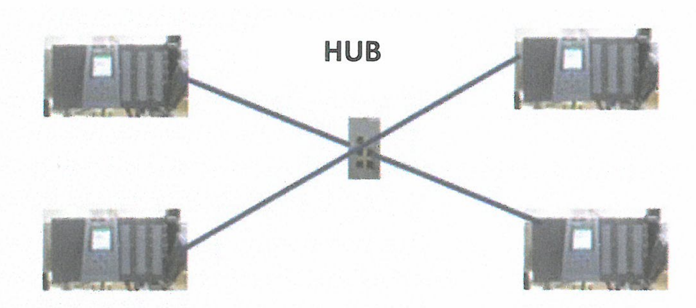


Figura 17

Puentes (*bridges*)
Los puentes comunican información entre dos segmentos distintos (diferentes grupos de red). La información pasa de un segmento a otro a través del puente. Si la comunicación se realiza entre dispositivos del mismo segmento, la información no pasa por el puente. Un puente comprueba la dirección de destino y hace copia hacia el otro segmento si allí se encuentra la estación de destino. La principal diferencia de un repetidor y un *hub* se halla en que este pasa toda la información al otro segmento, esté o no el dispositivo (figura 18).

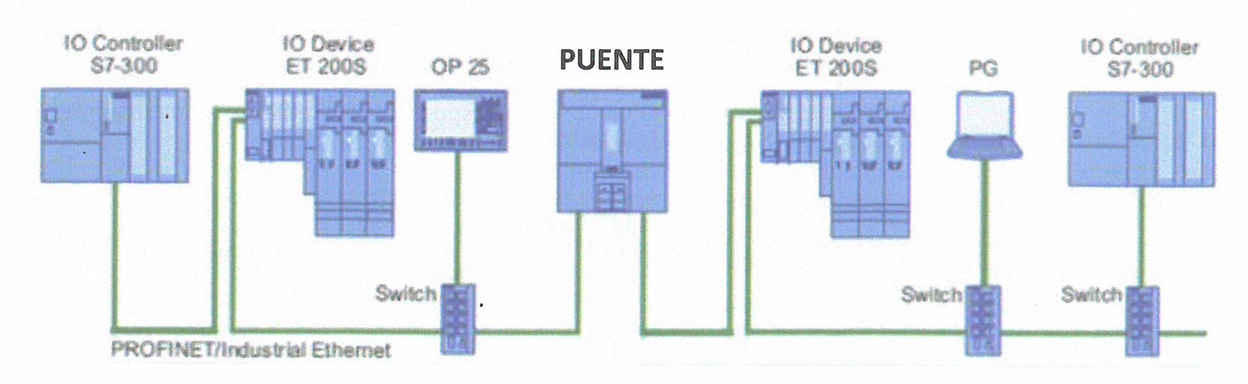


Figura 18

***Switch* (conmutador)**
El *switch* es similar al *hub*; la diferencia se encuentra en que el *switch* cuenta con un nivel superior de decisión. No envía la información a todos los dispositivos conectados. Realiza una selección y solo lo envía al nodo al que va dirigido. Igualmente, también interconecta dos o más segmentos de red, pero, en este caso, del mismo grupo de red.

Hoy solo se utilizan *switch* en vez de *hub* (figura 19).

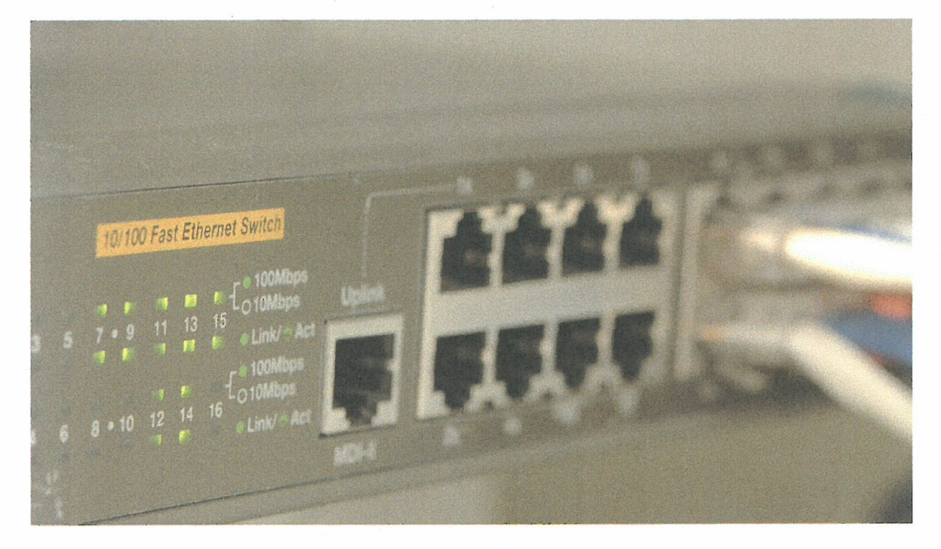


Figura 19

Encaminadores (*routers*)
Operan entre redes aisladas que utilizan protocolos similares (por ejemplo, TCP/IP) y direcciones o encaminan la información de acuerdo con la mejor ruta posible (figura 20).

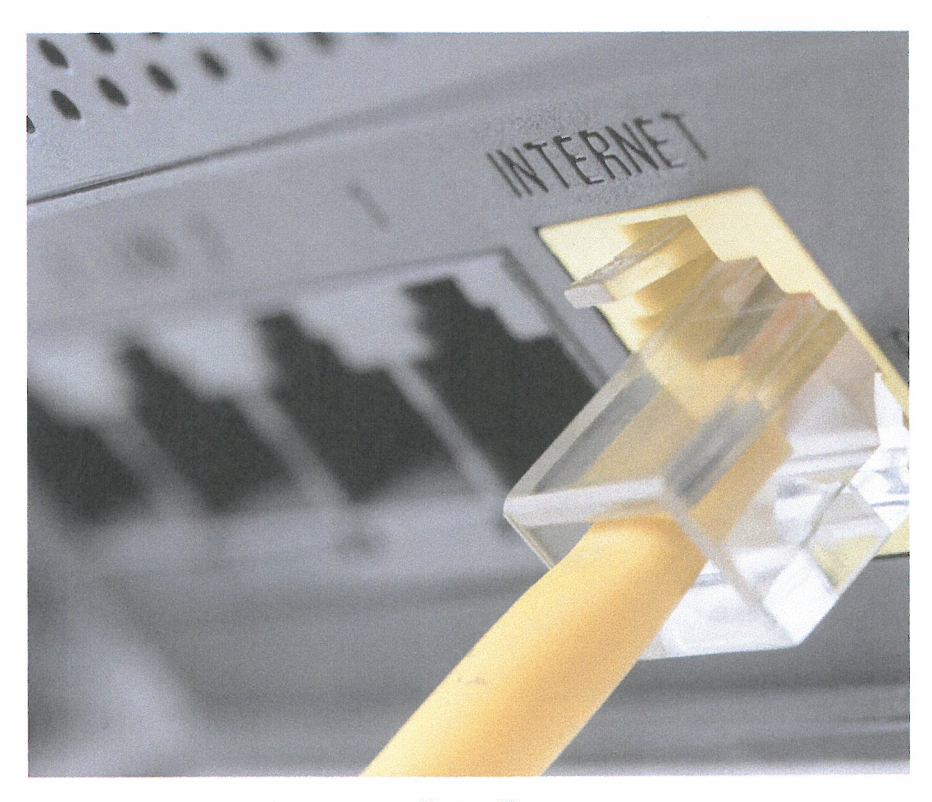


Figura 20

Recuerda . . .

Para interconectar redes, se necesita utilizar dispositivos que hagan posible que dichas redes puedan ser escalables, es decir, que puedan ampliarse en función de las necesidades del momento.

La primera función de un *router* es conocer si el destinatario de un paquete de información se halla en nuestra propia red o en una remota. El *router* encamina los datos hacia la red de destino si se encuentra en su grupo; si no fuera así, conectará con otros *routers* para realizar la comunicación.

El ejemplo más claro es la conexión a Internet que se dispone en las casas y que sale a través de un *router*.

Existen *routers* que son también *switch* con cuatro puertos y punto de acceso wifi.

Pasarela (*Gateway*)

Las pasarelas son traductores de red; más bien, de sus protocolos. Una pasarela comunica redes distintas, de protocolos diferentes. Por ejemplo, una pasarela puede conectar un segmento Profibus con otro segmento Profinet.

Los *gateways* deben desensamblar las tramas y paquetes que le llegan para obtener el mensaje original y, a partir de este, volver a reconfigurar los paquetes y las tramas, pero de acuerdo con el protocolo de la red donde se encuentra la estación de destino (figura 21).

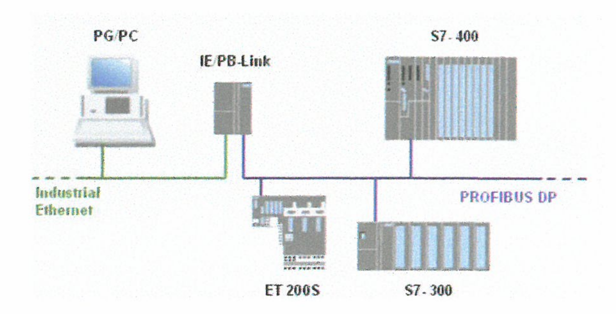


Figura 21

Unidad 2 Comunicaciones industriales

En esta unidad veremos:

Recuerda . . .

En los sistemas de comunicación industrial, las redes más instaladas son Profibus y Profinet. Actualmente, por el número de nodos colocados, el líder es Profinet.

2.1 Introducción

En este libro se tratan las comunicaciones industriales desde el punto de vista más práctico. Solo se utiliza aquella teoría absolutamente necesaria como conocimiento o por resultar imprescindible para la aplicación. Por ello, en este tema se abarcan las comunicaciones más importantes y de una manera muy resumida.

Se van a estudiar las redes Profibus, Profinet y AS-i. Actualmente, la red que se está extendiendo y que destaca entre todas es Profinet.

Una vez realizada la introducción de estas redes, se pasará al diseño y configuración de diversos ejercicios de cada una de ellas y su interacción entre sí.

Nos encontramos en pleno desarrollo de lo que se ha venido a llamar la Industria 4.0 y las comunicaciones industriales desempeñan un papel importante. La Industria 4.0 tiene por objetivo valorar la próxima generación de los procesos de producción que relacionan el mundo digital de los sensores, actuadores y procesadores con la tecnología de las comunicaciones. De este modo, se unirán el mundo virtual, Internet y la nube con el mundo real de la producción.

Profibus, Profinet y AS-i constituyen tres buses de comunicación basados en sistemas abiertos, es decir, sin propietario. Fue entre los años 1987 y 1990 cuando se creó Profibus y este fue el primer bus importante que lanzó el desarrollo de las comunicaciones industriales. Un grupo de empresas alemanas participaron en el primer lanzamiento: Bosch, Klockner Moeller y Siemens, además de ABB, AEG, Honeywell, Landis+Gyr, Phoenix Contact, Rheinmetall, RMP, Sauter-cumulus y Schleicher.

En el año 1973 se desarrolló la red Ethernet, pero no es, hasta finales de la década de los ochenta, cuando Ethernet empieza a tomar gran impulso. En aquellos tiempos, la industria se hallaba muy alejada de este tipo de redes. En 1985, Ethernet fue introducida en la industria como Industrial Ethernet por SIEMENS AG, con el nombre de «Sinec H1». A principios del siglo XXI, el grupo de empresas que lideró Profibus creó Profinet, que aunaba las ventajas obtenidas con Profibus con las de la red Ethernet.

El objetivo de dichas redes era acceder a los dispositivos que se encontraban cerca de la máquina de una forma más eficiente, de modo que se redujeran costes y se distribuyeran las tareas entre los procesadores de la información.

En 1990, en Alemania, un grupo de compañías de éxito creó un sistema de bus para conectar sensores y actuadores. Lo llamaron Actuator Sensor Interfaz (AS-i).

El sistema AS-i surgió como complementación al resto de sistemas ya existentes y la meta principal era aumentar la velocidad de conexión entre sensores, actuadores y sus controladores.

2.2 Profibus

Profibus es un bus de campo abierto que se utiliza en las aplicaciones más bajas de la pirámide de automatización, proceso y campo. Es un estándar que sigue la norma UNE IEC 61158 y las normas internacionales IEC 61784. Profibus puede utilizarse para aplicaciones críticas de alta velocidad y tareas de comunicación complejas y para transmitir pequeñas y medianas cantidades de datos entre los dispositivos que participan en la red.

Hoy día, este bus se encuentra en una clara decadencia al ser superado por la red de Profinet. Pero no se puede decir que Profibus haya muerto, porque hay muchos sistemas que lo siguen utilizando y aún se sigue instalando.

Algunas de las características de este bus son:

✓ En cuanto al medio físico de transporte de la información:

- Cable a dos hilos trenzados y/o apantallados (impedancia característica de 150 Ω).



- Fibras ópticas de vidrio y plástico.
- Sistemas de transmisión inalámbrica.
- Velocidad de red configurada desde 9,6 kbit/s a 12 Mbit/s.
- Conexión a la red de un máximo de 127 equipos. No obstante, el número máximo de equipos que pueden estar activos al mismo tiempo es de 32.
- Longitud máxima de la red variable, en función del medio físico utilizado: hasta 9 km con medio eléctrico y hasta 90 km si se utiliza fibra óptica.

Se disponen de tres tipos de Profibus (FMS, DP y PA):

✓ FMS (**E**specificación de **M**ensajes de **C**ampo): ha sido diseñado para la comunicación de autómatas en pequeñas células de red, unos con otros, y para la comunicación con elementos Profibus DP. Las aplicaciones típicas son PLC comunicados entre ellos, que controlan zonas donde existen otros PLC, y otros dispositivos de E/S.

✓ Profibus DP (**P**eriferia **D**escentralizada): se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas; es decir, no enlazan redes de PLC sino de dispositivos de entrada y salida.

Por lo general, las redes Profibus DP incorporan **un maestro** y varios esclavos. La configuración del maestro le permite reconocer qué tipos de esclavos se hallan conectados, así como sus respectivas direcciones. El maestro inicializa la red y verifica si los esclavos coinciden con la configuración. Continuamente, el maestro escribe los datos de salida en los esclavos y lee desde allí los datos de entrada. Una vez que un maestro DP ha configurado correctamente un esclavo, este último le pertenece. Si aparece otro maestro en la red, tendrá un acceso realmente limitado a los esclavos del primer maestro. Un maestro se encarga de «tomar» la red; es decir, toma el mando y decide cuándo y con quién comunicar.

✓ Profibus PA (**A**utomatización de **P**rocesos): es la comunicación compatible, adicional a Profibus DP, con la tecnología de transmisión, que permite a los usuarios ir a un área explosiva y de alto riesgo.

La tecnología de transmisión de Profibus PA se corresponde con el estándar internacional IEC 1158-2; para DP y FMS, se encuentra el estándar IEC 61158, que coincide, en el aspecto eléctrico, con el estándar RS485.

Se puede decir que Profibus FMS es para aplicaciones multimaestro y las aplicaciones DP y PA son monomaestro.

Profibus FMS emplea el método de acceso al medio de *token bus* (bus con paso de testigo), mientras que Profibus DP y PA utiliza el método de acceso al medio por sondeo (*polling*).

2.2.1 Maestros y esclavos

Un maestro es siempre una estación activa, un PLC. En Profibus DP, el maestro se encarga de controlar la comunicación. En DP solo puede existir un maestro que gobierne la red. En la práctica, pueden hallarse más PLC en una red de Profibus DP, pero no podrán ejercer de maestro; solo uno será el maestro y los demás actuarán como esclavos con capacidad de procesar información.

Los esclavos son unidades de entrada y salida descentralizada; normalmente, se trata de dispositivos sin ninguna capacidad de decisión, no procesan información. Si es una entrada, tan solo se ponen a uno o cero, dependiendo de si el sensor se encuentra o no activo, pero no se puede pasar la información hasta que el maestro dé la orden. Hoy día existen estaciones descentralizadas con poder de decisión, resultan más «inteligentes» que una estación esclava pura.

En la figura 22 se puede ver un esclavo de la marca Siemens. Este fabricante los llama ET 200. Hay ET dedicadas para entradas/salidas concretas y en función del tipo de ET 200, Siemens le añade un número. En la figura 22 puede verse una ET 200S.

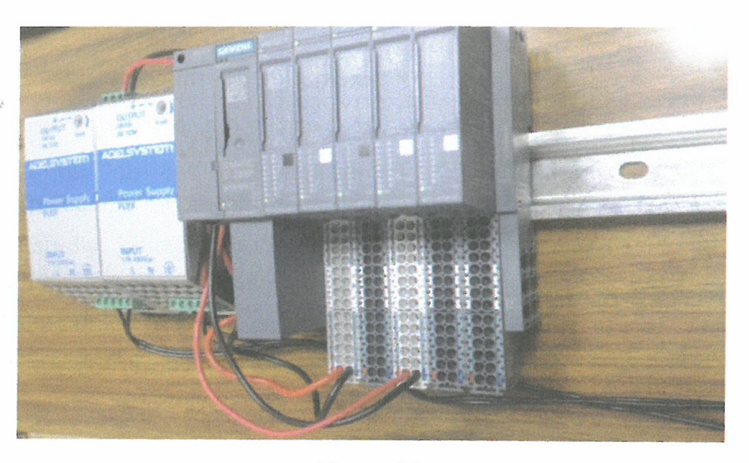


Figura 22

2.2.2 Cableado y conectores

La topología típica para Profibus es la de bus. Para este tipo de redes se deben utilizar resistencias terminales en los nodos que hacen de principio y final; es decir, los PLC o ET que solo tienen conexión por un lado han de disponer de una resistencia que compense la red por el margen que no tiene nada conectado. Si, por ejemplo, son tres PLC, el del centro se conecta a otro PLC por cada lado; sin embargo, los PLC de los extremos solo lo hacen con un PLC y, por ello, deberá disponerse de una carga por el otro lado. Los conectores de Profibus ya llevan dicha resistencia con un conmutador para poder conectar o desconectar la resistencia terminal. En la figura 23 se muestra uno de estos conectores con el conmutador para la resistencia.

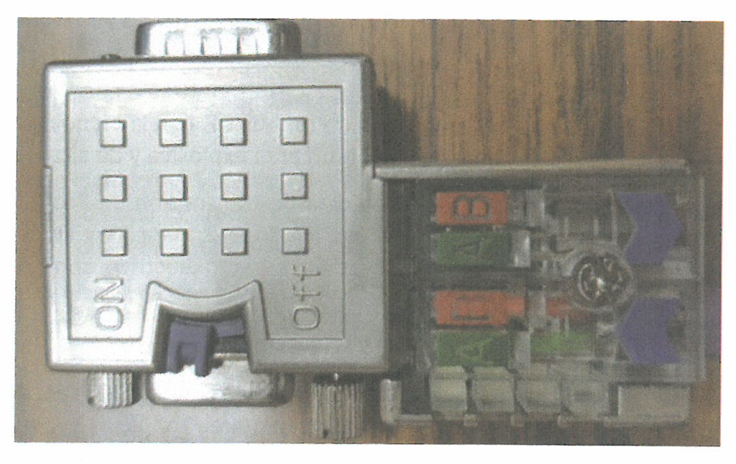


Figura 23

El cable para Profibus DP y FMS es de color morado en par trenzado apantallado, de cobre, con diferentes recubrimientos según el ambiente. En la figura 24 se observan dos tipos de cables con diferente apantallado.

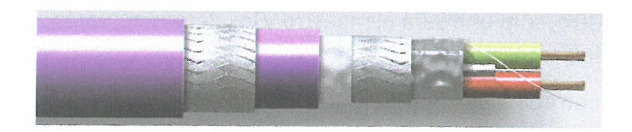




Figura 24

Los conectores son derivadores que disponen de una boca para un cable de entrada y otra para la salida hacia otro dispositivo. En el caso de ser un extremo, solo llevará cable de entrada y la resistencia se pondrá en ON para el otro extremo, tal como se aprecia en la figura 25.

Recuerda . . .

Profibus es un tipo de comunicación industrial que tiene sus aplicaciones en la parte más baja de la pirámide de automatización, es decir, en la zona de proceso y de campo.

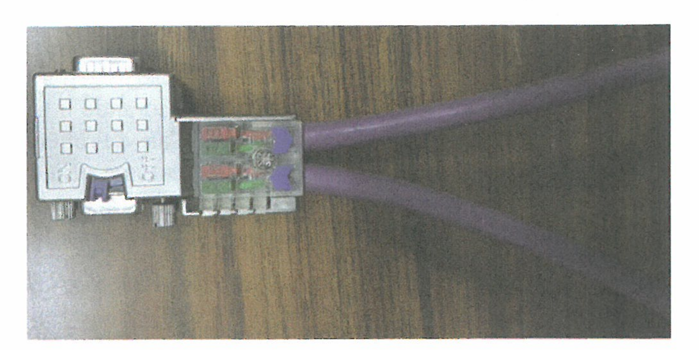


Figura 25

Para pelar el cable, existen pelacables que facilitan la labor (figura 26).

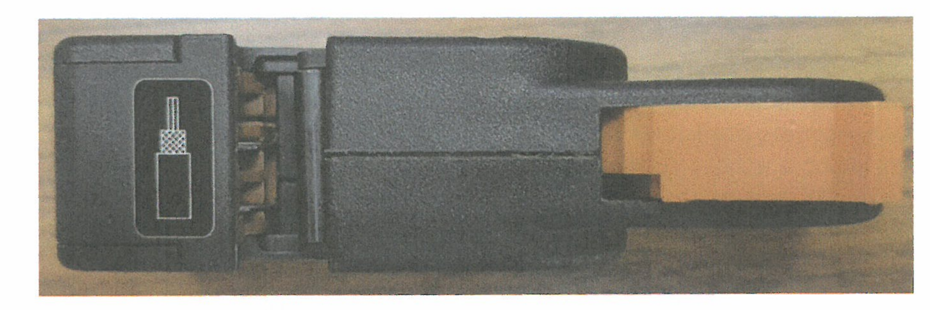


Figura 26

En la parte reservada a los ejercicios prácticos, encontrará otros aspectos importantes para la configuración de redes Profibus.

2.3 Profinet

Profinet nace como una continuación de Profibus; adquiere la experiencia con Profibus y las ventajas de las redes de Ethernet. Se considera el auténtico bus en tiempo real. Mucho tiempo antes de aparecer Profinet, ya existía una red muy parecida que se llamaba Ethernet Industrial. De hecho, Profinet constituye una variante de Ethernet Industrial. Pero no fue, hasta la aparición de Profinet, cuando estas redes alcanzaron el éxito. El motivo de que Ethernet Industrial no tuviera implantación en la industria se basa en dos aspectos fundamentales: uno de ellos era la imposibilidad de disponer de unidades de E/S descentralizadas, al igual que ocurría en Profibus; el otro motivo reside en el propio principio de funcionamiento de Ethernet respecto al método de acceso al medio. Faltaba el determinismo, es decir, saber con seguridad un rango de tiempo en el que se iba a producir el envío de información. Existen muchas aplicaciones donde este detalle no resulta importante, pero hay otras donde es fundamental para su correcto funcionamiento.

Profinet solucionó estos dos problemas. El más complicado era la falta de determinismo. Por tal motivo,Profinet creó una prioridad para los datos manejados por él mismo. Se debe tener presente que, en estas redes, al conectarse a Ethernet, se manejan datos de tipo industrial pero también de tipo ofimático. Con la creación de esta «autopista» reservada a los datos industriales, se puede establecer un tiempo de comunicación.

Con la solución del tiempo real, vino la creación de dispositivos de E/S descentralizados. Actualmente existen muchas estaciones descentralizadas de Profinet. Siemens las sigue llamando ET 200; incluso algunos módulos de E/S resultan compatibles para colocar también en estaciones de Profibus; lo único que cambia es el tipo de interfaz de comunicación, lógicamente.

Con lo dicho anteriormente, se puede definir Profinet como el estándar abierto de Ethernet Industrial de la asociación Profibus Internacional (PI), según la norma IEC 61784-2. Constituye el estándar de comunicación actualmente más utilizado en redes de automatización.

Partiendo de una conectividad básica, como es el cable de Ethernet, y unas tramas de comunicaciones establecidas, que corresponderían a los niveles 1 y 2 del modelo OSI, Profinet va incorporando nuevas funcionalidades denominadas «perfiles» de utilidad, como ProfiSafe o ProfiEnergy, mediante una interpretación específica para cada caso de los datos transmitidos, modificando el nivel 7 (de aplicación). En el caso de ProfiSafe, se transmiten datos de seguridad (*safety*) y, en el caso de ProfiEnergy, datos y comandos para el ahorro y control energético.

Profinet emplea el método CSMA/CD como control de acceso al medio.

Con Profinet se pueden conectar dispositivos, sistemas y conjuntos de dispositivos aislados entre sí, lo que mejora tanto la velocidad como la seguridad de sus comunicaciones, reduce costes y optimiza la producción. Por sus características, Profinet permite la compatibilidad con comunicaciones Ethernet más propias de entornos TI, al aprovechar todas las características de estas, salvo la diferencia de velocidad que posee una comunicación Ethernet situada en una red corporativa frente al rendimiento en tiempo real que necesita una red industrial. Con el uso de Profinet, la integración y la comunicación total se muestran más sencillas por el hecho de utilizar Ethernet.

Además, Profinet abre inmensas posibilidades de comunicación con las tecnologías de información (IT). Conectar un PLC a Internet y disponer de un sencillo diagnóstico resulta tan fácil como poner su dirección en un explorador de Internet. Igualmente enviar *e-mails* y prestar teleservicio es mucho más sencillo con las redes Profinet.

Adicionalmente, el uso del estándar Profinet en el nivel E/S proporciona las siguientes ventajas:

- Mejora de la escalabilidad en las infraestructuras.
- Acceso a los dispositivos de campo a través de la red. Profinet, al utilizar Ethernet en su comunicación, facilita acceder a dispositivos de campo desde otras redes de una forma más fácil.
- Ejecución de tareas de mantenimiento y prestación de servicio desde cualquier lugar. Permite acceder a dispositivos de campo mediante conexiones seguras como, por ejemplo, VPN, para realizar mantenimientos remotos.

2.3.1 Tipos de comunicación de Profinet

Profinet utiliza tres servicios de comunicación:

- **Standard TCP/IP:** este servicio se utiliza para funciones no deterministas, como parametrización, transmisiones de vídeo/audio y transferencia de datos a sistemas TI de nivel superior.
- **Real Time:** las capas TCP/IP no se emplean para ofrecer un rendimiento determinista a las aplicaciones de automatización, pues funcionan con unos tiempos de retardo en el rango de 1-10 ms. Este hecho representa una solución basada en *software* adecuada para aplicaciones típicas de E/S; se incluyen control de movimiento y requisitos de alto rendimiento.
- **Isochronous Real Time:** la priorización de señal y la conmutación programada proporcionan una sincronización de alta precisión para aplicaciones como el control de movimiento. Las velocidades de ciclo en rangos de submilisegundos son posibles, con *jitter* (variabilidad temporal durante el envío de señales digitales) en el rango de submicrosegundos.

Existen varios protocolos definidos dentro del contexto Profinet. Se exponen alguno de estos protocolos junto con su uso concreto:

- **Profinet/CBA:** protocolo asociado a las aplicaciones de automatización distribuida en entornos industriales.
- **Profinet/IO:** a veces llamado Profinet-RT (RealTime), se utiliza para comunicaciones con periferias descentralizadas. Se emplea en las aplicaciones que se verán en este libro.
- **Profinet/IRT:** transferencia de datos isócrono en tiempo real.

Recuerda . . .

Profinet es la red industrial más extendida actualmente. El protocolo más usado es el Profinet I/O. Lo que hace útil y flexible a Profinet reside en basarse en el estándar Ethernet.

Recuerda . . .

Si hay que destacar algún inconveniente de la red de Profinet, ese es el de la ciberseguridad. Exponerse al exterior, mediante Internet, hace que los avances de Profinet vayan ligados a los de la ciberseguridad.

2.3.2 Seguridad

De la accesibilidad proporcionada por Profinet se encarga un protocolo muy expuesto a Internet, por lo que se debe mejorar la ciberseguridad de las redes en las que se despliega.

Algunas de las buenas prácticas para asegurar entornos industriales que usan Profinet son las siguientes:

- Protección frente a errores, funcionamientos incorrectos y manejo adecuado de los incidentes que puedan originarse con procedimientos previamente creados.
- Prevención frente a accesos no autorizados que producirían manipulaciones en la red o espionaje.
- Uso de estándares y dispositivos de seguridad probados y certificados (cortafuegos, VPN, IDS/IPS, etcétera).
- Medidas en la infraestructura de red: las arquitecturas de redes planas simplifican y facilitan la comunicación entre sistemas y dispositivos. Sin embargo, estas presentan un reto para mantener la disponibilidad, la estabilidad y la seguridad de la red, ya que un atacante con acceso a la red podría acceder a todos los nodos. La segmentación de red con uso de VLAN, *router*, cortafuegos, etc., contribuye, de manera significativa, a mitigar tales problemas.

Posible uso de VLAN para asegurar una red con Profinet

- Protección de los dispositivos finales: gracias a la desactivación selectiva de servicios; la desinstalación de aplicaciones innecesarias o la modificación de contraseñas por defecto, podemos asegurar los dispositivos finales presentes en una red industrial si se minimizan los puntos débiles y las brechas de seguridad.

2.3.3 Identificación de una red Profinet

Una red de Profinet se define por los mismos parámetros que la red Ethernet, ya que cumple con su estándar.

Los dispositivos que se conecten a una red de Profinet deberán diferenciarse del resto. En Profibus, era suficiente con la asignación de una dirección. En Profinet, no será el único parámetro que se deba definir.

La dirección será una IP (Internet Protocolo) con su máscara de subred. Además, todos los dispositivos Profinet presentarán otra dirección denominada MAC. Por último, deberán disponer de un nombre.

Dirección IP y máscara de subred

La dirección IP se halla formada por cuatro grupos de números de 8 bits (*byte*) cada uno. Estos grupos pueden ser desde 0 a 255.

Esa dirección IP está formada por dos partes: la dirección del dispositivo (nodo) y la dirección del grupo de red a la que ese dispositivo se conecta. Para identificar esas dos partes: se dispone de la máscara de subred.

La máscara de subred está formada también por cuatro grupos de números de 8 bits cada uno, pero, en este caso, cada bit solo puede ser uno o cero.

En el siguiente ejemplo se aclararán estos dos conceptos. Supongamos la siguiente dirección IP 10.0.127.100 y la máscara de subred 255.255.255.0. En la tabla siguiente se desglosa cada dirección.

	Dirección de red			D. de Nodo
IP	10.	0.	127.	100
Máscara de subred	255.	255.	255.	0

La máscara de subred indica qué parte de la dirección IP es la dirección de la red y qué parte es la dirección del dispositivo (nodo). Los bits de la máscara de subred correspondientes a unos indican, en su *byte* parejo de la IP, que es dirección de grupo de red y los que sean cero, que es dirección de nodo. En este ejemplo, la dirección de red es 10.0.127 y la del nodo, 100. Los dispositivos conectados a redes con diferentes direcciones no se pueden comunicar entre ellos. En este ejemplo se pueden conectar 256 dispositivos a la misma red, desde la dirección 0 a la 255.

Esto sería otro ejemplo: IP :192.168.34.71 y subred 255.255.0.0, donde la parte de dirección de red es 192.168 y la de nodo, 34.71.

MAC (Media Access Control)
Otro elemento identificador en una red de Profinet es la MAC (Media Access Control), que es la dirección física del dispositivo. Está formada por 6 bloques de 2 caracteres hexadecimales (48 bits, 6 bloques de 4 bits cada uno). Esta dirección es única y ningún otro aparato la puede llevar. Se encuentra serigrafiada en el propio dispositivo. Esta dirección la determina y configura el IEEE (los últimos 24 bits) y el fabricante (primeros 24 bits) a través del Oganizationally Unique Identifier.

La mitad de los bits de una dirección MAC identifican al fabricante y la otra mitad, al dispositivo. Concretamente, las tres primeras parejas identifican al fabricante y la otra mitad, al aparato.

Ejemplo de MAC   0A:7B:C2:9D:7E:4B

Nombre de dispositivo (Device)
Por último, todos los dispositivos de Profinet deben disponer de un nombre cualquiera que se identifique en la red y que sea único. Ha de ser puesto por el técnico cuando realiza la configuración.

Cableado y conectores

Profinet puede utilizar cualquier cable utilizado para una red Ethernet. En Ethernet, el más utilizado es el cable UTP *(unshielded twisted pair)* o par trenzado no apantallado (figura 27).

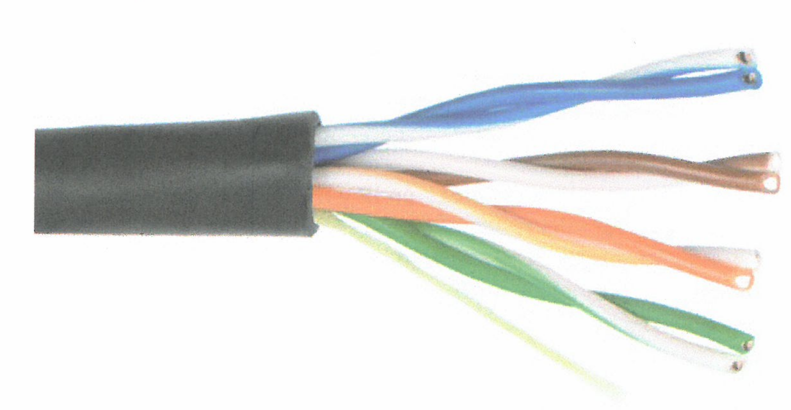


Figura 27



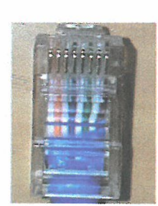
Figura 28

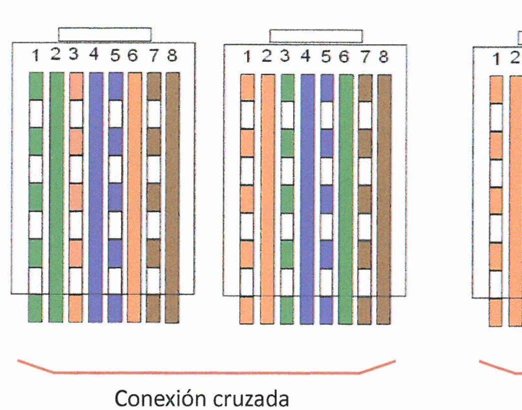
El conector que se utiliza es el RJ45 (figura 28).

El conexionado de estos cables en los conectores dependerá de si se requiere un cable directo o cruzado. El cable directo es aquel que tiene sus dos conectores con los cables puestos en el mismo orden. El cable cruzado es aquel que, entre un conector y otro, muestra intercambiados los cables de recepción y envío.

La necesidad de utilizar uno u otro depende del tipo de aplicación. Si, por ejemplo, se desea conectar dos PLC (dos nodos cualesquiera), se debe utilizar un cable cruzado. Si la conexión se realiza a través de *switch*, no resulta necesario ya que el propio dispositivo detecta el tipo de cable y realiza la conexión.

En la figura 29 se puede ver la disposición de las dos conexiones.





Conexión cruzada   Conexión directa

Figura 29

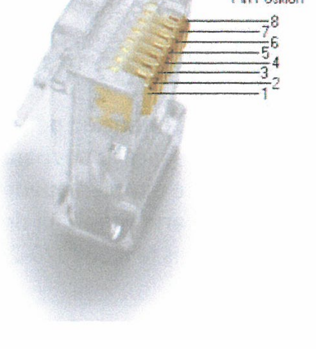


Figura 30

En esta figura 30 muestra la numeración de los pines del conector RJ45.

El cable utilizado en Profinet difiere del de Ethernet en el apantallamiento. Se emplean en las industrias donde el ambiente eléctrico resulta mucho más hostil que en una oficina. En las industrias, las interferencias abundan. Hay diferencia entre los dos tipos de cables, el de Profinet y el de Ethernet normal. En la figura 31 puede verse un cable utilizado en Profinet con los apantallamientos adecuados para su uso en la industria.

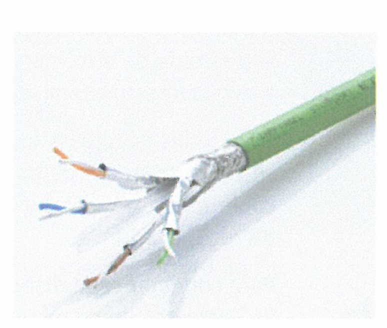


Figura 31

Aunque cualquier cable de Ethernet valdría, el utilizado en Profinet puede tener solo 4 conductores, tal como se muestra en la figura 32. En este caso solo se usan las señales de transmisión y recepción.

Los colores corresponden a las siguientes señales:

SEÑALES	COLORES
Dato de transmisión +	Amarillo
Dato de transmisión –	Naranja
Dato de recepción +	Blanco
Dato de recepción –	Azul

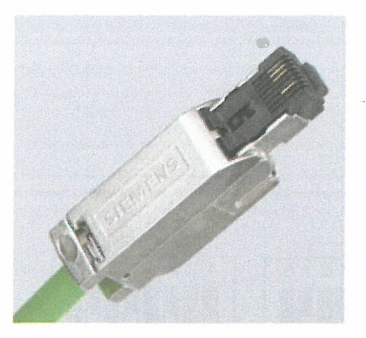


Figura 33

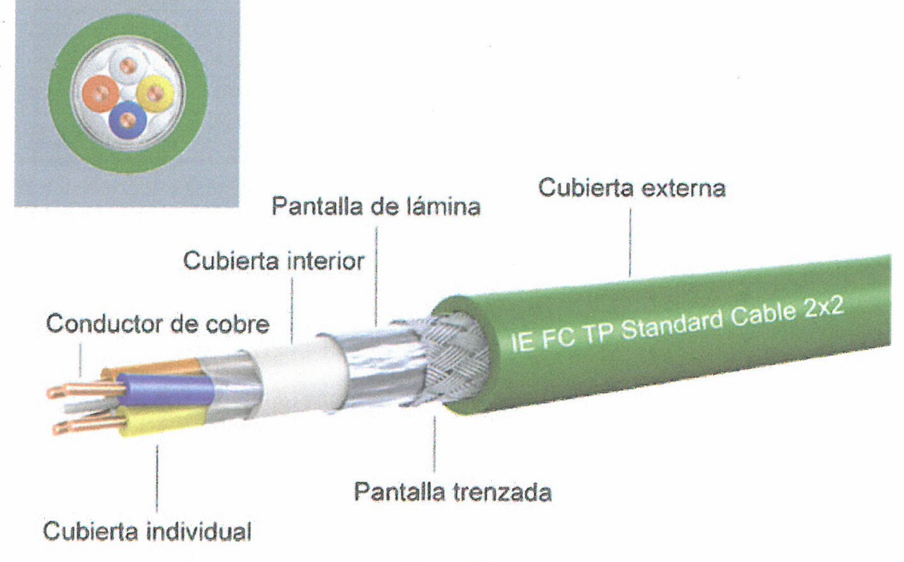


Figura 32

El conector que se utiliza en Profinet tampoco es igual al empleado en Ethernet. Este resulta mucho más robusto, con el fin de poder trabajar en ambientes industriales (figura 33).

2.4 AS-Interface (Actuator Sensor Interface). AS-i

AS-i es un bus de comunicación abierto que se creó en 1990 con la intención de interconectar los sensores y actuadores con un solo cable bifilar. Las aplicaciones para las que estaba creada esta red eran digitales, aunque actualmente también puede transmitir señales analógicas. La red de AS-i se acomoda en la parte más baja de la pirámide de automatización. La norma que rige este estándar es la IEC62026-2, a escala internacional, y la EN 50295, en Europa.

Una de las características de esta sencilla red reside en que, por el mismo cable se transmiten los datos y también la alimentación de los esclavos de AS-i. Este cable está formado por los dos conductores de cobre y una cubierta de goma de recuperación.

Para conectar cada dispositivo, se realiza mediante la perforación del aislamiento. Se llama «cable de recuperación» porque ,una vez eliminada dicha perforación, el cable recupera su estado.

El cable es de color amarillo y dispone de una protección mecánica mediante un escalón para impedir que se coloque con la polaridad cambiada. En la figura 34 se observa el escalón y la forma de conexión entre el módulo y el cable.

Figura 34

Características AS-i

Algunas de las características de la comunicación AS-i son:

- El AS-Interface está optimizado para la conexión de sensores y actuadores binarios y analógicos.
- A través del cable de AS-i, se realiza tanto el intercambio de datos entre sensores/actuadores (esclavos de AS-i) y el maestro AS-i como también la alimentación de corriente de los sensores y actuadores.
- Se compone de un cableado sencillo y económico y un montaje fácil con técnica de conexión por desplazamiento del aislamiento y muestra una gran flexibilidad gracias al cableado de tipo árbol.
- El maestro AS-i necesita, para el intercambio de datos cíclico con participación, de hasta 31 estaciones, un máximo de 5 ms (para 62 estaciones, 10 ms).

- Las estaciones (esclavos de AS-i) conectadas al cable de AS-i pueden ser sensores/actuadores con conexión AS-i integrada o bien módulos AS-i, a los que se conectan, en cada caso, hasta ocho sensores/actuadores binarios convencionales. Con módulos de AS-i estándar se pueden operar hasta 124 actuadores y 124 sensores en el cable AS-i.
- Si se utilizan módulos AS-i con área de direcciones ampliadas, se operan hasta 248 actuadores y 248 sensores en un maestro extendido.

Montaje de un módulo de AS-i

El montaje de un módulo de AS-i conectado al cable de AS-i resulta particularmente sencillo gracias a la técnica de conexión descrita.

El módulo de acoplamiento se atornilla o se encaja en un riel de perfil de sombrero de 35 mm. Se coloca el cable de AS-i, se enclava en una guía por encima de las cuchillas de contacto y se fija mecánicamente. Con esto no se establece contacto eléctrico. Al atornillar el módulo de usuario, se presiona el cable de AS-i contra las cuchillas de contacto. Estas penetran en los conductores en dos puntos y garantizan una conexión segura. Si el montaje se realiza correctamente, el módulo posee el grado de protección IP67 (figura 35).

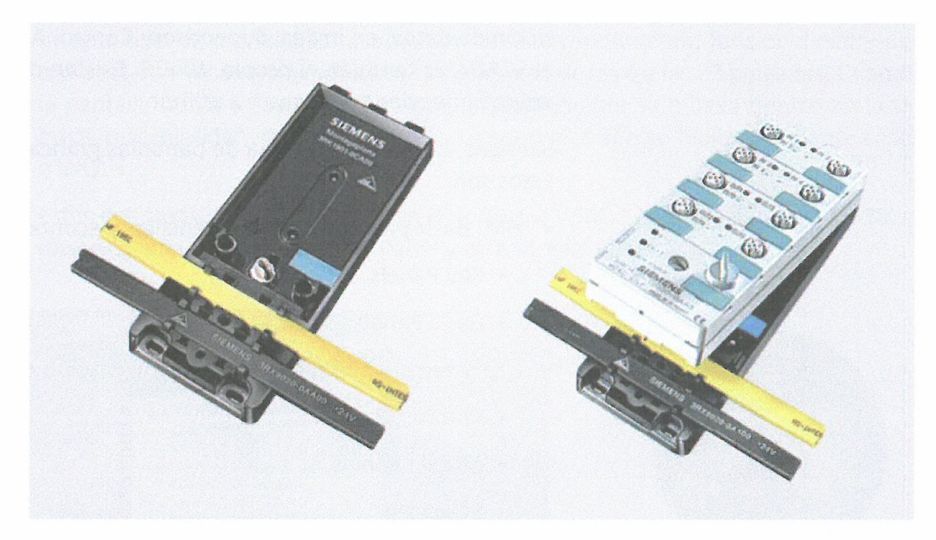


Figura 35

Elementos necesarios

Para formar una red de AS-i, se necesita contar con una fuente de alimentación AS-i, esclavos y un maestro.

Las fuentes de alimentación de AS-i generan un voltaje de 30 V CC, con una gran estabilidad y reducida ondulación. Estas alimentan a los equipos electrónicos de la red, que consiste en los módulos de AS-i, y al maestro, así como a los sensores conectados. Por medio del desacoplamiento integrado de los datos, las fuentes de alimentación aseguran la separación de datos y energía, transmitidos al mismo tiempo a través del cable de dos hilos de AS-i.

Los esclavos son módulos sensores/actuadores que llevan el interfaz de AS-i; también pueden ser módulos de aplicación general de E/S a los que se conectan sensores estándar que no sean AS-i.

El maestro se encarga de realizar la comunicación de los esclavos de la red de AS-i y de intercambiar esos datos con otro bus de comunicación como Profibus o Profinet. Este es el procedimiento más normal de utilización. En una red de AS-i, solo puede haber un maestro.

Todos los esclavos y el maestro deberán presentar una dirección de AS-i que se debe colocar mediante una programadora. En la figura 36 puede verse la programadora.



Figura 36



2.5 HMI y WinCC

Los sistemas de información y control gráficos suponen una parte importante en los sistemas de producción. De poco servirían las automatizaciones si no se pudiera obtener información de lo que está sucediendo y, más aún, si no introdujeran variaciones en el sistema de forma externa al propio programa. Por este motivo, las pantallas, los SCADAS, son un elemento primordial en el esquema de un sistema automático. Para decidir un tipo de pantalla, no solo se debe encontrar aquella que se adecua a la aplicación; resulta mucho más importante que la solución encontrada sea flexible y con futuro, que se pueda integrar en redes superiores y que se muestre capaz de responder a las crecientes demandas de transparencia y provisión de datos.

Los dispositivos *hardware* (las pantallas) que realizan la función de visualización de la producción industrial se encuadran en lo que se denomina HMI, es decir, Interfaz Humano Máquina. Todas estas pantallas deben programarse o configurarse mediante un *software* adecuado. En el caso de Siemens, ese *software* se denomina WinCC. Dentro de los dispositivos de visualización y control, se encuentran también los SCADAS. Un SCADA es un *software* destinado a que el ordenador actúe como un dispositivo más dentro de la cadena de control de un sistema automático. El acrónimo SCADA viene de Sistemas de Supervisión, Control y Adquisición de datos; en inglés, **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition. En Siemens, el SCADA es también el propio WinCC. Existen diversos WinCC, en función de las aplicaciones donde se vayan a utilizar.

Siemens, dentro de su gama de pantallas gráficas HMI, dispone de varios niveles. Estos son:

- ✓ HMI Basic para aplicaciones sencillas y económicas:
  - Key Panels.
  - Basic Panels.
- ✓ HMI Avanzado, mediante la implementación de aplicaciones complejas:
  - Comfort Panels.
  - Mobile Panels.
  - PC-based.

Existe un *software* WinCC para cada tipo de pantalla. Los tipos de WinCC son:

- ✓ WinCC Basic.
- ✓ WinCC Comfort.
- ✓ WinCC Advanced.
- ✓ WinCC Professional.

Las pantallas Basic Panels requieren el *software* mínimo de WinCC: el WinCC Basic. Estas pantallas resultan sencillas para capacidades funcionales limitadas. Las hay de 4 a 12 pulgadas. Funcionan a partir de un manejo combinado de pantalla táctil y teclas de libre configuración.

Dentro de este grupo se encuentran los Key Panels, pequeños teclados con los que se implementan para paneles de mando.

En la figura 37 se muestran un Key Panel y un Basic Panel.

Recuerda . . .

Las pantallas de visualización son los nexos de unión entre las máquinas y las personas. Por ello, resultan imprescindibles en la fábrica moderna para actuar sobre el sistema industrial, sin necesidad de modificar los programas.



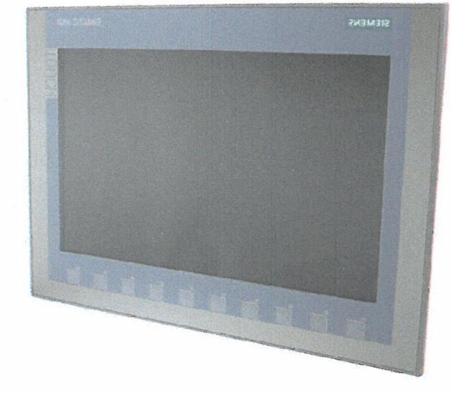


Figura 37

Dentro de las pantallas del grupo HMI Advanced, se encuentran los paneles Comfort, los Mobile y los basados en PC. Los Comfort Panels se han concebido para aplicaciones de visualización potentes a pie de máquina. Las características de este tipo de pantalla se resumen en un alto rendimiento, funcionalidad y numerosas interfaces integradas. Los tamaños van de las 4 a las 22 pulgadas. Cuando la pantalla no va a estar fija, sino que probablemente se mueva dentro de la fábrica, recomiendan los Mobile Panels. La comunicación se realiza mediante cable o WLAN.

Estos dos tipos de pantallas utilizan el *software* WinCC Comfort. En la figura 38 se observan una pantalla Comfort Panel y otra Mobile Panel.





Figura 38

En el grupo de paneles Advanced, también se encuadran los paneles basados en PC. Estos son PC industriales, y unen la eficacia y potencia de un PC con la seguridad y robustez necesarias para el manejo dentro de la industria. Sus tamaños oscilan entre 7 y 22 pulgadas.

Estos tipos de pantallas requieren el *software* WinCC Advanced.

En la figura 39 puede verse un panel basado en PC.



Figura 39

Por último, se encuentra el WinCC Profesional; este *software* es el SCADA de Siemens. Antiguamente solo la denominación WinCC pertenecía al SCADA y los programas para las pantallas se denominaban WinCC flex. Actualmente, se ha simplificado mucho en cuanto a los complementos necesarios para las aplicaciones de ingeniería. Ahora se han integrado dentro de los paquetes de *software* generales.

En la figura 40 se expone un resumen de los paneles de Siemens y el *software* requerido.

SIMATIC WinCC (TIA Portal)

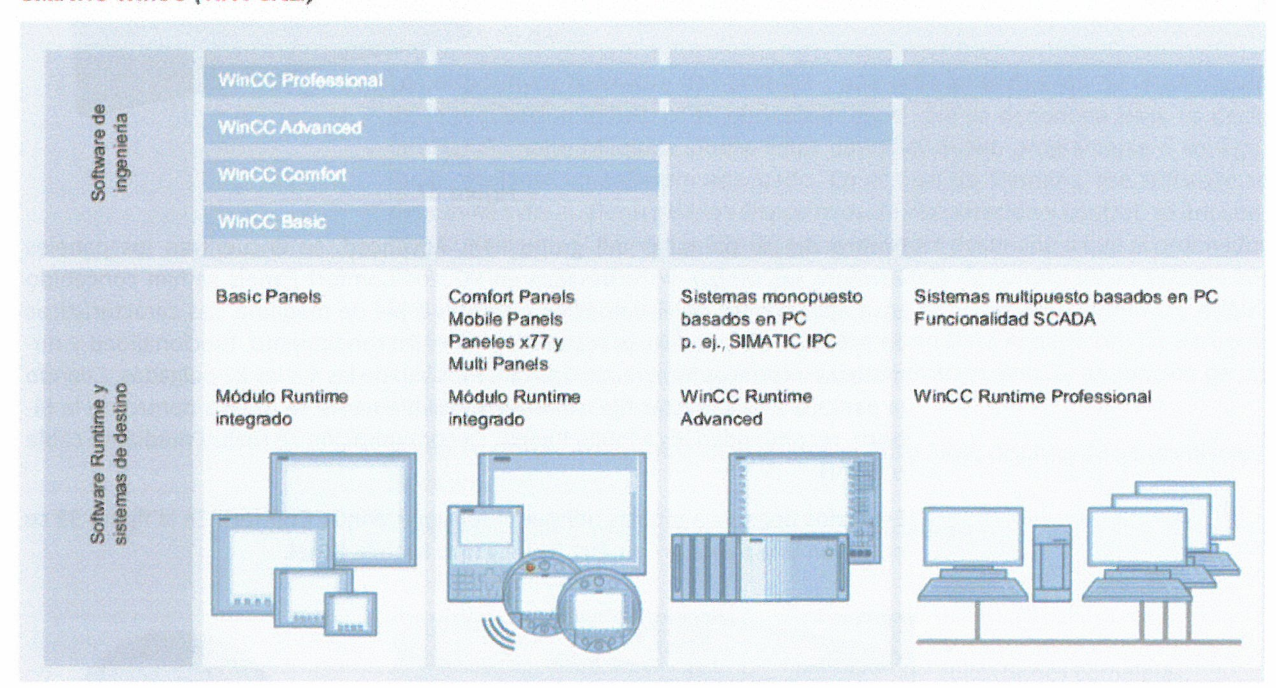


Figura 40

Unidad 3 Ejercicios prácticos de Profibus

En esta unidad veremos:

1. Ejercicio de Profibus PLC + ET
2. Ejercicio de Profibus PLC maestro + PLC esclavo + ET

1. Ejercicio de Profibus PLC + ET

Enunciado

Se trata de realizar una comunicación entre un PLC y una ET, vía Profibus. El PLC será el que controle a la ET 200. La comunicación se realizará mediante Profibus DP.

Material necesario

Se va a utilizar material enumerado a continuación:

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Un PC-Adapter USB A2:
   - Una estación de trabajo descentralizada ET 200M que lleva los siguientes módulos:
   - Un módulo de 16 entradas y salidas digitales de 24 V.
   - Un módulo de 4 entradas y 2 salidas analógicas.
   - Un cable Profibus para las conexiones.

Realización

El proceso de realización será el explicado a continuación:

En primer lugar, se deberá crear el proyecto en TIA PORTAL con el PLC utilizado. A continuación, se ha de colocar en el proyecto la estación de trabajo descentralizada de que se disponga. Desde la *Vista de redes*, se accede al catálogo en la carpeta *Periferia descentralizada*. En la figura 41 se indica el resultado de esta situación.

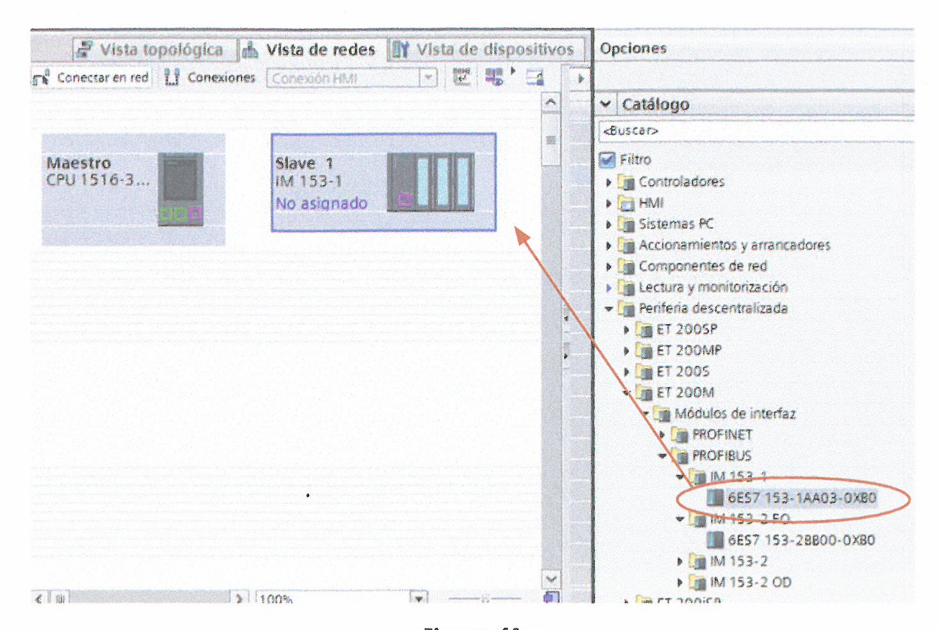


Figura 41

Esta ET 200M requiere añadir los módulos de E/S que tenga. Si se accede a la vista de dispositivos de la ET (tocando con el ratón dos veces en la ET), aparecerá en el catálogo los módulos disponibles para esta ET. Véase la figura 42.

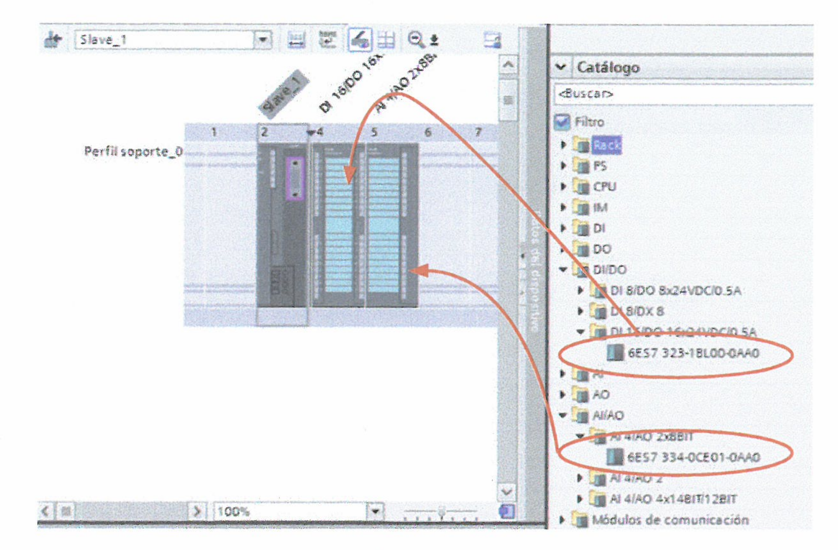


Figura 42

El siguiente paso es unir el PLC y la ET mediante una conexión Profibus. Para ello basta con tocar con el ratón en la conexión de Profibus del PLC (conexión morada) y llevarlo hasta la ET, como se puede ver en las figuras 43 y 44.

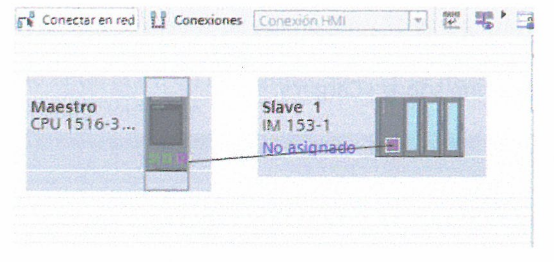


Figura 43



Figura 44

Al realizar la conexión, a la ET se le habrá asignado una dirección de Profibus. Todos los dispositivos de la red deben contar con una dirección distinta. Se debe tener en cuenta que la dirección de la ET ha de mostrar en el proyecto la misma dirección que posee realmente. Todas las ET disponen de un procedimiento físico para poder poner una dirección de Profibus. Como ya se sabe, las direcciones de Profibus constituyen un número comprendido entre el 1 y 127.

En la figura 45 se representa la ET 200M, con unos *microswitches* para poner su dirección. En dicha figura se ve que la dirección real de esa ET es la 8. Si se cambia la dirección modificando el estado de los *microswitches* no hay que olvidarse de apagar y volver a encender la ET. De esta manera se recopilará la nueva dirección. Lo mejor es apagar la ET para cambiar la dirección.

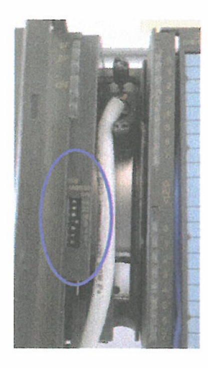


Figura 45

Ahora se debe modificar la dirección que le ha asignado el proyecto de TIA PORTAL. Para eso, se ha de ir a la vista de dispositivos y, tocando en la interfaz de Profibus, se accede a las propiedades. Desde allí se puede modificar la dirección. Se aprecia en la figura 46.

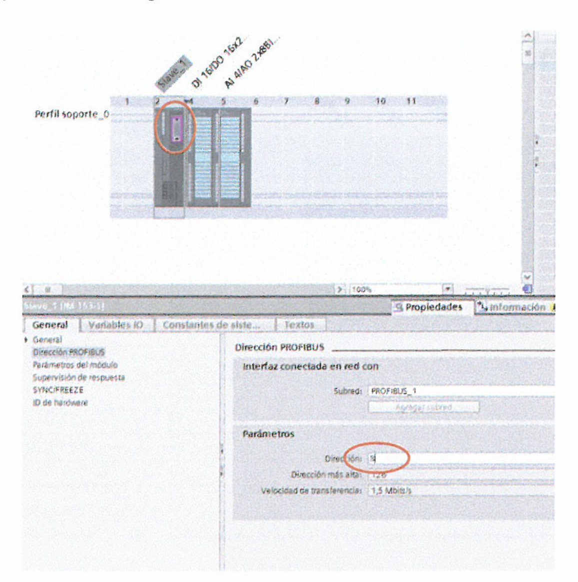


Figura 46

Ha llegado el momento de unir la red físicamente mediante un cable Profibus y pinchar encima del conector de Profibus: el PC-Adapter. Una vez hecho esto, solo queda enviar la configuración al PLC. Antes de poder utilizar la interfaz de Profibus, se debe enviar la configuración a través de Profinet. Para ello, se accede a enviar mediante el icono . Saldrá la ventana indicada en las figuras 47 y 48, donde se deberá seleccionar el tipo de interfaz (PN/IE) y la tarjeta de red del ordenador. Pulse sobre «Iniciar búsqueda» y, posteriormente, cuando se haya encontrado el PLC y se active la opción, presione sobre «Cargar».

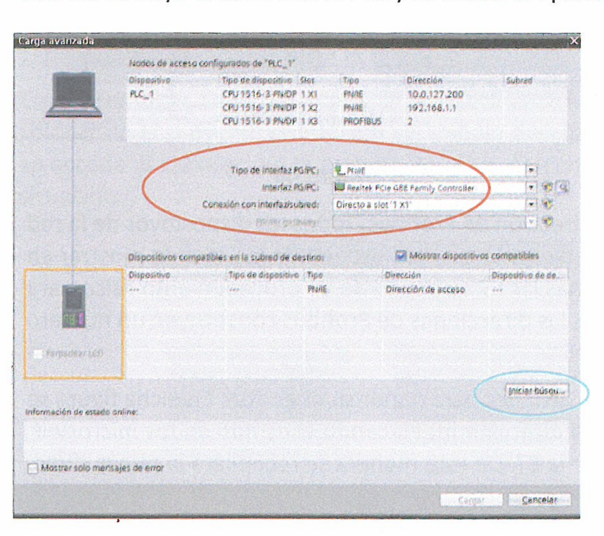


Figura 47

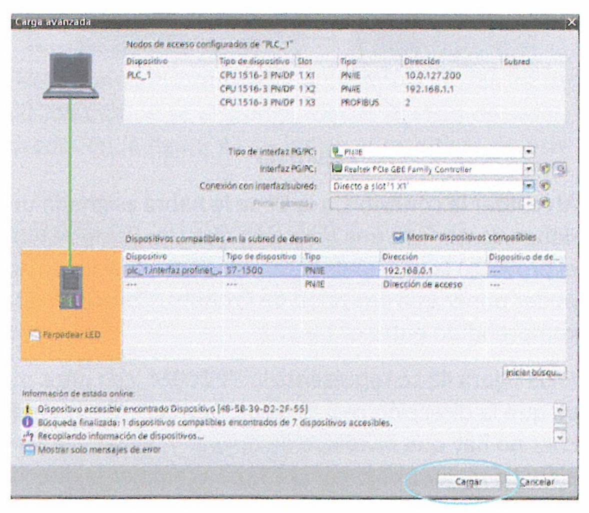


Figura 48

Una vez que se ha enviado vía Profinet, ya se puede utilizar el PC-Adapter vía Profibus. Para hacer la prueba de que se puede enviar sin problemas vía Profibus, debe repetirse la operación de carga pero esta vez vía Profibus. Se vuelve a pulsar en el icono de cargar en dispositivo. Si no sale la pantalla donde se cambia la interfaz, se debe de acceder a la carga mediante el menú *Online* y se selecciona la opción «*Carga avanzada en dispositivo...*». Entonces vuelve a salir la ventana anterior y se cambian las interfaces, como se indica en la figura 49. Es necesario tener conectado el PC-Adapter al PLC (conector Profibus) y al PC mediante USB.

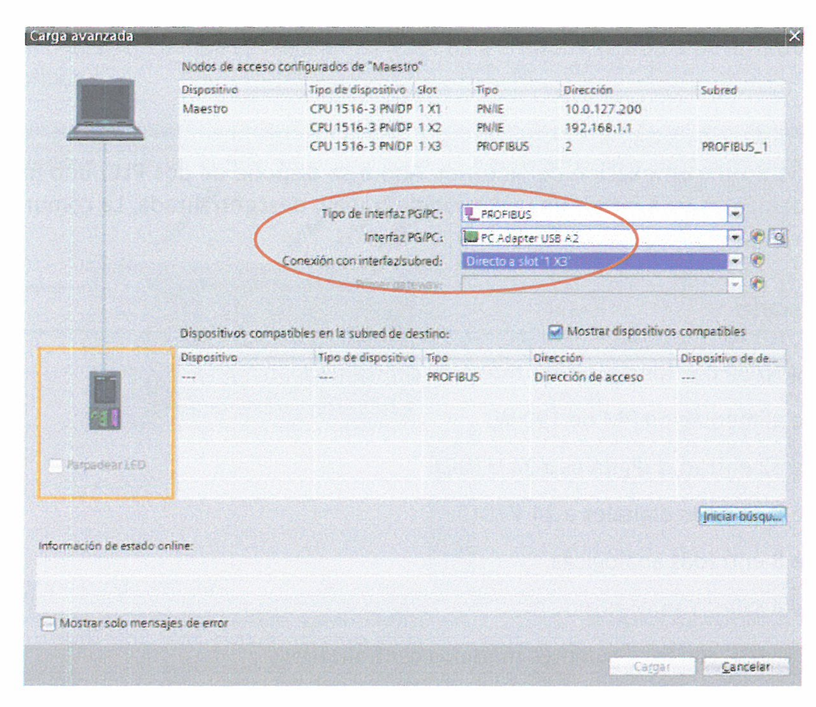


Figura 49

Una vez cargado vía Profibus, ya se habrá realizado la conexión entre el PLC y el ordenador, mediante el PC-Adapter, de forma que, si se accede a la red mediante TIA PORTAL, se puede «ver» qué hay en la red de Profibus, siempre que, al enviar, no se encuentre encendido en rojo el diodo led.

Para poder ejecutar el programa, se deben conocer las direcciones de E/S de la periferia, tanto la del PLC como la de la ET. En las figuras 50 y 51 se muestran estas direcciones para el ejercicio, para el PLC y para la ET, respectivamente.

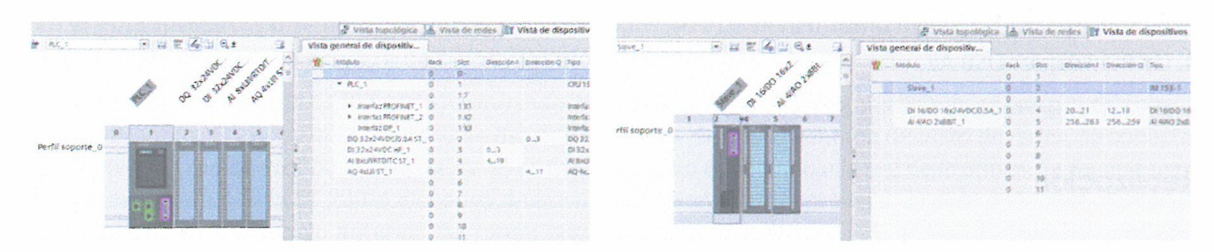


Figura 50 **Figura 51**

El programa para comprobar el comportamiento de la comunicación puede ser algo similar a este:

L EB0 // *Byte* de entrada del PLC

T AB12 // *Byte* de salida de la ET

L EB20 // *Byte* de entrada de la ET

T AB0 // *Byte* de salidas del PLC

2. Ejercicio de Profibus PLC maestro + PLC esclavo + ET

Enunciado

El siguiente ejercicio es una ampliación del anterior. Ahora se dispone de dos PLC: uno hará de maestro y otro de esclavo. Además, se va a incluir una estación de trabajo descentralizada. La comunicación será mediante Profibus DP.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una CPU 314C 2PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación SITOP de 5 A.
   - Un módulo de 16 entradas digitales a 24 V (DC).
   - Un módulo de 16 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas y 2 salidas analógicas.
3. Una estación de trabajo descentralizada ET 200M que lleva los siguientes módulos:
   - Un módulo de 16 entradas y salidas digitales de 24 V.
   - Un módulo de 4 entradas y 2 salidas analógicas.
4. Un PC-Adapter USB A2.
5. Un cable de Profibus para las conexiones.

Realización

Como se ha dicho, este caso es una continuación del anterior. Uno de los PLC será el maestro DP y el otro, esclavo DP. El PLC que hace de esclavo resulta algo más que una simple estación pasiva (ET 200), ya que cuenta con la capacidad de procesar información, aspecto que no tiene una ET normal. El PLC 1516 no puede hacer de esclavo porque no dispone de esa capacidad. Por lo tanto, será el PLC 314C el que deberá configurarse como esclavo. Al autómata que hace de esclavo se le puede llamar *esclavo inteligente*.

Con esta situación, el PLC esclavo no puede comunicarse con el PLC maestro de forma directa, como se ha visto en el ejercicio anterior entre el PLC y su ET.

En este caso habrá que utilizar unas áreas de E/S para intercambiar los datos entre los dos. Respecto a la ET 200, que se asignará al PLC maestro, no puede ser de otra forma, porque un esclavo (por muy PLC que sea) nunca poseerá y controlará a otro esclavo. Además, el único que se podrá comunicar directamente con él será el PLC maestro; por su parte, el PC esclavo deberá utilizar esa área de E/S para comunicarse con el PLC maestro y, con ello, con la ET.

Partiendo del ejercicio anterior, se debe añadir el nuevo PLC2 esclavo, que es un S7-314C. Como este PLC2 va a ser esclavo, se debe configurar como tal; para ello hay que dirigirse a sus propiedades desde la ventana de «Vista de dispositivos» y pulsar dos veces sobre el conector de Profibus. Para que se pueda pinchar el PC-Adapter en este PLC2, debe activarse la opción *Test, puesta en servicio y routing*. En la figura 52 se indica esta situación. También debe saber que este PLC S7-314C dispone de un conector para la interfaz de Profibus, aunque resulta igualmente válido para la comunicación MPI. Se puede seleccionar manualmente, pero no

se preocupe por ello ya que, cuando se haga la red y se una a Profibus, quedará de forma automática puesto como interfaz Profibus.

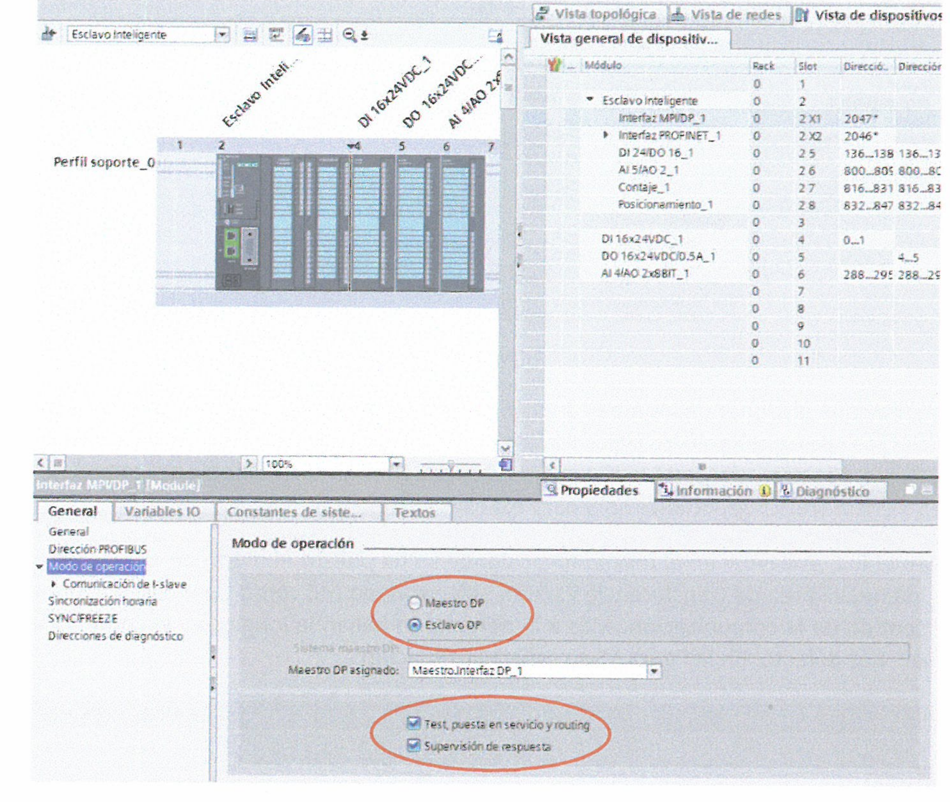


Figura 52

En el PLC1 maestro, no se debe hacer nada porque todos los PLC vienen como maestros por defecto. Además, en este caso, el PLC S7-1516 no posee la opción de ser esclavo de Profibus.

Una vez que se tiene el PLC1 (S7-1516) como maestro y el PLC2 (S7-314C) como esclavo, se unen mediante la red de Profibus. El PLC1 y la ET 200M ya se mostrarán unidos. Desde la ventana de *Vista de redes*, se conecta el esclavo a la red de Profibus. En la figura 53 se presenta todo conectado a la red de Profibus. Se ha de tener la precaución de que la ET 200M disponga de la misma dirección en el proyecto que en la ET real, como ya se ha hecho en el ejercicio anterior. De forma automática, al unir los dos PLC, el PLC esclavo ha quedado asignado al PLC maestro, pues no existe otra posibilidad. Si hubiera más maestros, se seleccionaría a quién se asigna el PLC esclavo.



Figura 53

Si hubiera que cambiar alguna dirección de Profibus en el PLC, se accedería a las propiedades de la conexión y allí se modificaría (figura 54).

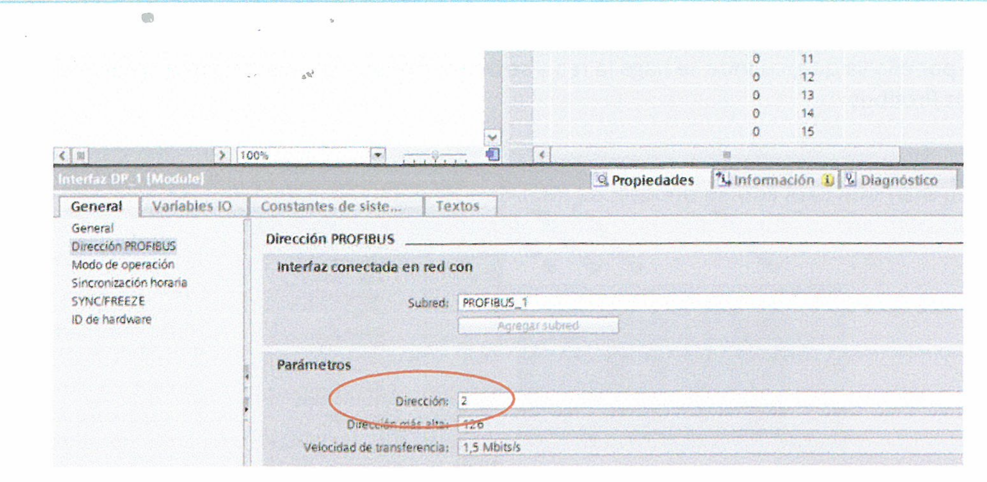


Figura 54

Seguidamente, hay que disponer de las áreas de comunicación entre el PLC maestro y el PLC esclavo. Así, de una forma cíclica, se intercambiarán los datos ambos de PLC. A esas direcciones se las denomina *direcciones virtuales* de E/S del maestro y del esclavo. Estas direcciones deben situarse dentro del campo de direccionamiento de E/S y encontrarse disponibles solo para esta situación.

Esto se hace en el PLC esclavo y en el mismo lugar donde se ha puesto el modo de operación (más abajo), tras agregar una nueva área de transferencia y haber seleccionado ME como tipo. Tocando sobre **la flecha,** se cambia el sentido de la comunicación. Más a la derecha se hallan la longitud y la coherencia; se puede dejar lo que sale por defecto. En la figura 55 se refleja la situación.

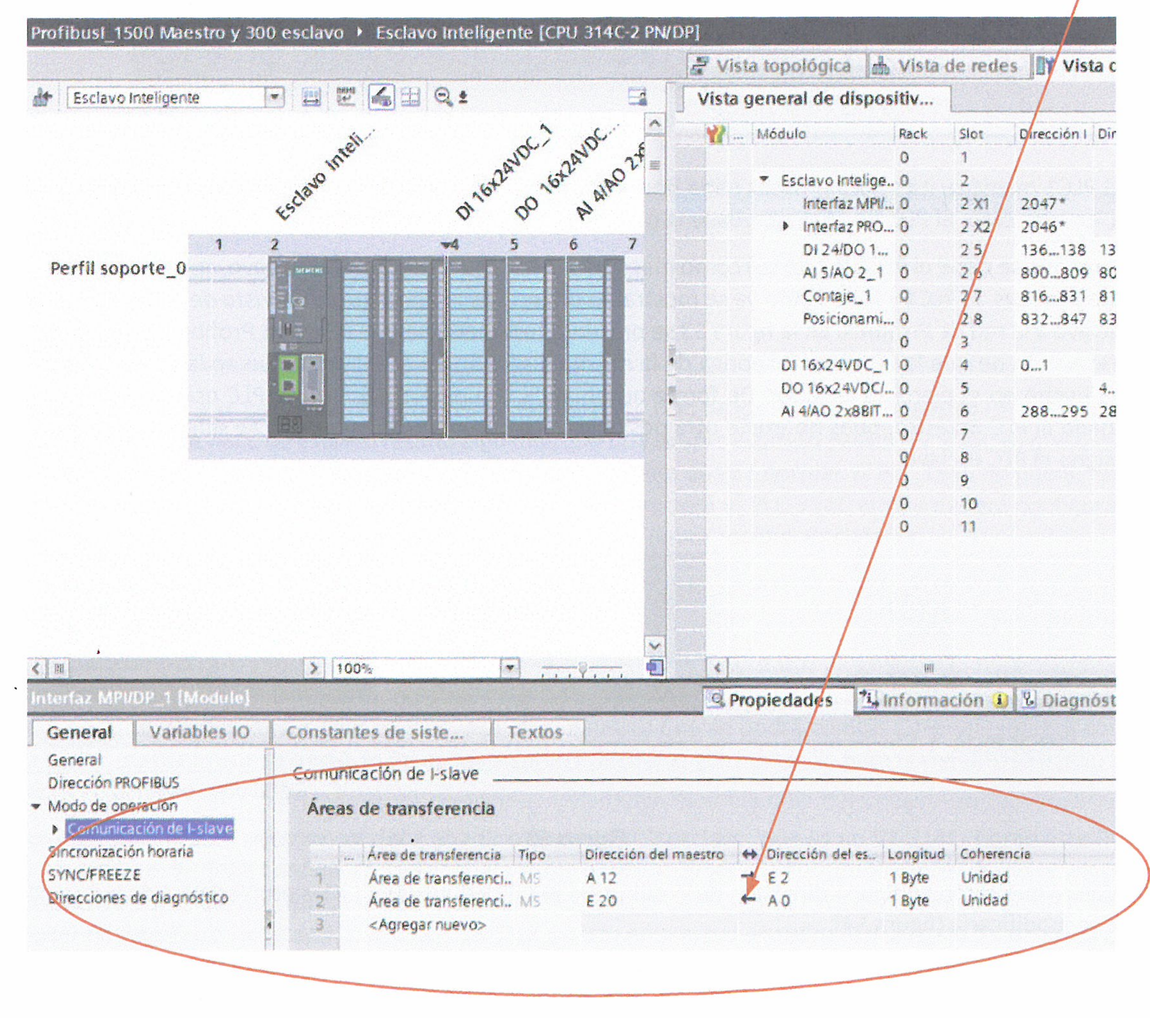


Figura 55

Por último, se envía la configuración de cada PLC de **forma INDIVIDUAL** a través de PC-Adapter USB A2, según se indica en la figura 56. Una vez transmitido todo, deberían desaparecer los ledes de indicación de fallo, tanto de SF como de BUS y pasar de STOP a RUN cada PLC para rearrancar.

Si se produjera algún error, intente cargar los OB de error siguientes: OB82 *(Diagnostic error interrupt)*, OB6 *(Rack or station failure)*, OB121 *(Programming error)* y OB122 *(IO access error)*. No escriba nada en ellos; tan solo agréguelos en el proyecto del PLC maestro y envíelos al PLC.

En el caso de que siguiera habiendo errores, utilice el acceso *online* y observe dónde indica el error.

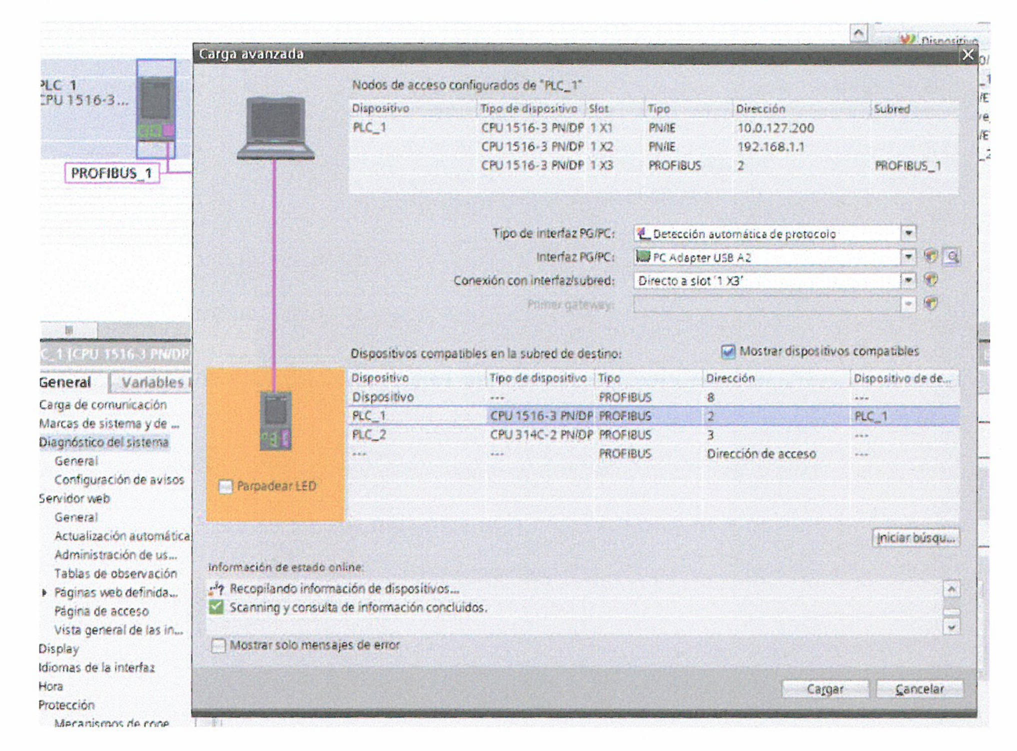
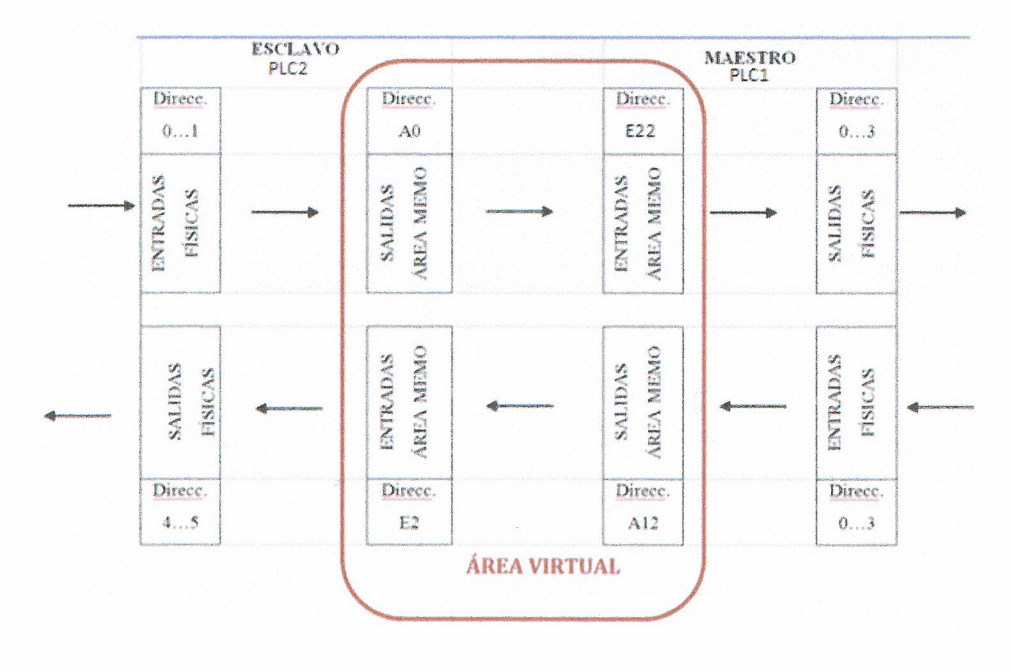


Figura 56

Ahora solo queda confeccionar un programa para que todos los dispositivos puedan comunicarse entre sí. No debe olvidar que la ET 200M pertenece al maestro y no se puede acceder desde el PLC esclavo directamente a ella, aunque sí a través de las direcciones de E/S virtuales creadas. También necesita conocer las direcciones físicas de los PLC. En la siguiente tabla se reflejan todas esas direcciones, para el caso de este ejercicio.



Los OB1 pueden ser como estos:

OB1 del MAESTRO (PLC1)

```
L   "Entrada_PLC1"                %EB0
T   "SalidaPLC1_hacia PLC2"       %AB12
T   "Salida_ET200M"               %AB14

L   "EntradaPLC1_desde PLC2"      %EB22
T   "Salida_PLC1"                 %AB0
T   "Salida_ ET200M"              %AB13
```

OB1 del ESCLAVO (PLC2)

```
L   "Entrada_PLC2"                %EB0
T   "Salida PLC2_hacia PLC1"      %AB0

L   "Entrada PLC1_ desde PLC2"    %EB2
T   "Salida_PLC2"                 %AB4
```

Las direcciones de la ET 200M, en este ejemplo, son el *byte* 13 y el 14, tanto para entrada como para salida.

Unidad 4 Ejercicios prácticos de Profinet



En esta unidad veremos:

3. Ejercicio de Profinet PLC + ET
4. Ejercicio de Profinet PLC + ET en sistema abierto
5. Ejercicio de Profinet de dos PLC y dos ET
6. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada
7. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada, con CP
8. Ejercicio de AS-i/DP con un PLC S7-1500
9. Ejercicio de dos PLC S7-1500 en comunicación abierta. Enlace TCP
10. Ejercicio de comunicación de PLC S7-1500, S7-1200, S7-300 y periferia
11. Ejercicio de integración de redes Profibus, Profinet y AS-i
12. Ejercicio de acoplamiento de dos redes de distintas subredes

Los OB1 pueden ser como estos:

OB1 del MAESTRO (PLC1)

```
L    "Entrada_PLC1"                 %EB0
T    "SalidaPLC1_hacia PLC2"        %AB12
T    "Salida_ET200M"                %AB14

L    "EntradaPLC1_desde PLC2"       %EB22
T    "Salida_PLC1"                  %AB0
T    "Salida_ ET200M"               %AB13
```

OB1 del ESCLAVO (PLC2)

```
L    "Entrada_PLC2"                 %EB0
T    "Salida PLC2_hacia PLC1"       %AB0

L    "Entrada PLC1_ desde PLC2"     %EB2
T    "Salida_PLC2"                  %AB4
```

Las direcciones de la ET 200M, en este ejemplo, son el *byte* 13 y el 14, tanto para entrada como para salida.

Unidad 4 Ejercicios prácticos de Profinet



En esta unidad veremos:

3. Ejercicio de Profinet PLC + ET
4. Ejercicio de Profinet PLC + ET en sistema abierto
5. Ejercicio de Profinet de dos PLC y dos ET
6. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada
7. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada, con CP
8. Ejercicio de AS-i/DP con un PLC S7-1500
9. Ejercicio de dos PLC S7-1500 en comunicación abierta. Enlace TCP
10. Ejercicio de comunicación de PLC S7-1500, S7-1200, S7-300 y periferia
11. Ejercicio de integración de redes Profibus, Profinet y AS-i
12. Ejercicio de acoplamiento de dos redes de distintas subredes

3. Ejercicio de Profinet PLC + ET

Enunciado

Se trata de realizar una comunicación entre un PLC y una ET. El PLC será el que controle a la ET 200.

Material necesario

Se va a utilizar el siguiente material:

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una estación de trabajo descentralizada ET 200SP que lleva los siguientes módulos:
   - Dos módulos de 16 salidas digitales de 24 V.
   - Dos módulos de 16 entradas digitales de 24 V.
3. Un cable de Profinet para las conexiones.

Realización

El proceso de realización será el mostrado a continuación:

Se deberá crear el proyecto en TIA PORTAL con el PLC utilizado. Una vez que se tiene el proyecto, se accede a la *Vista de proyecto*, a la pestaña de *Vista de redes* y se procede a añadir la ET. Se quedará como indica la figura 57. Desde allí y accediendo al catálogo, a la pestaña *Periferia descentralizada,* se busca la ET 200SP. En este caso, la referencia es 6ES7 155-6AU00-0BN0 y la versión, V1.1.0. Hay que fijarse en la versión que se coloca en el proyecto para que sea la misma que la disponible físicamente.

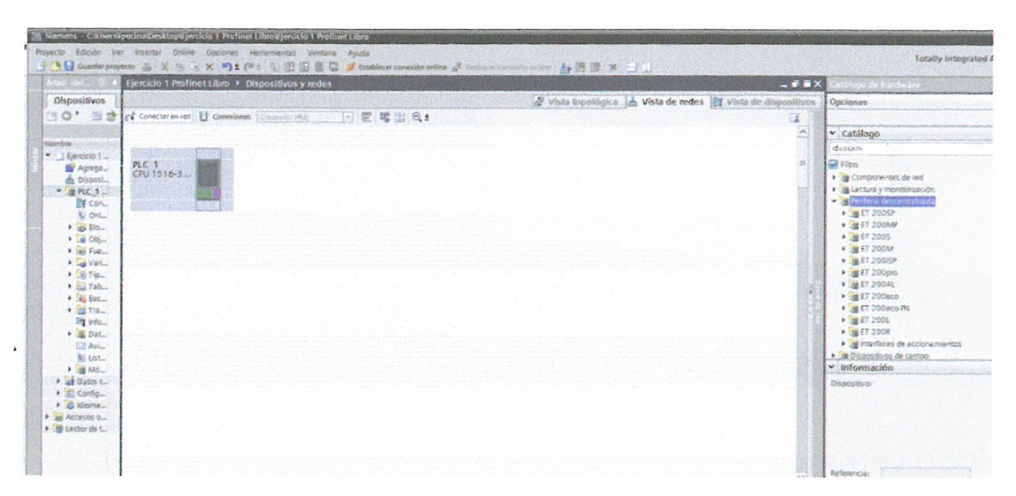


Figura 57

En la figura 58 se observa el lugar donde se encuentra la ET y dónde se puede comprobar su versión.

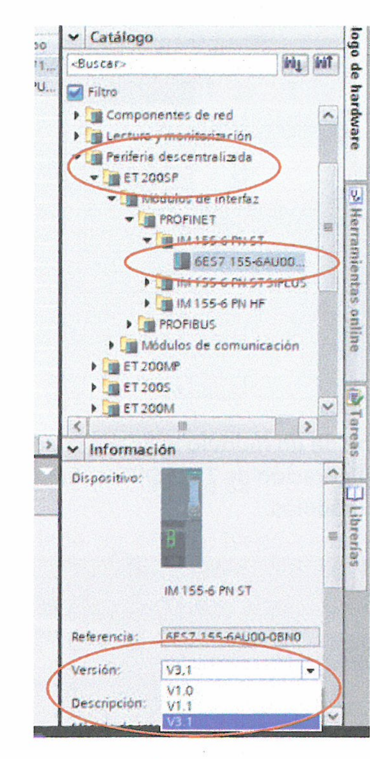


Figura 58

Particularidades de la ET 200SP

Cada estación de trabajo posee unas particularidades que hace que su configuración sea diferente a otras. En todas las estaciones de trabajo se debe diferenciar entre los módulos de borneado y aquellos correspondientes a los de entrada y salida.

En esta ET, los bloques de bornes de conexión son de dos tipos y se representan en la figura 59.

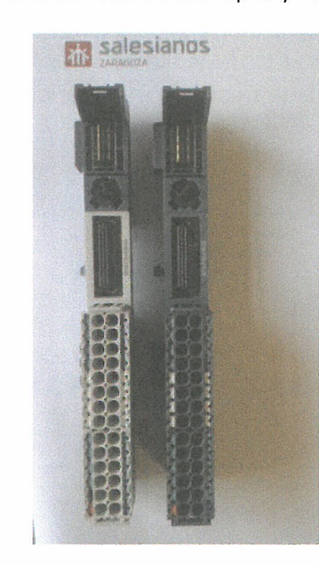


Figura 59

Existen bloques de borneado blancos y grises. Los blancos son los módulos de adaptación de alimentación y todos los bloques que se encuentren a la derecha se alimentarán con ese módulo y su fuente correspondiente. Estos bloques a la derecha del blanco serán grises. Se podría utilizar otra fuente de alimentación diferente para alimentar otros módulos de forma que, si fallara una alimentación, los otros módulos no se verían afectados. En la figura 60 se distinguen dos módulos blancos. Los módulos que se sitúan a la derecha se alimentan de una fuente diferente. Sobre estos bloques de borneado, se puede colocar cualquier módulo de E/S. Hay que fijarse igualmente en la versión de cada módulo de E/S.

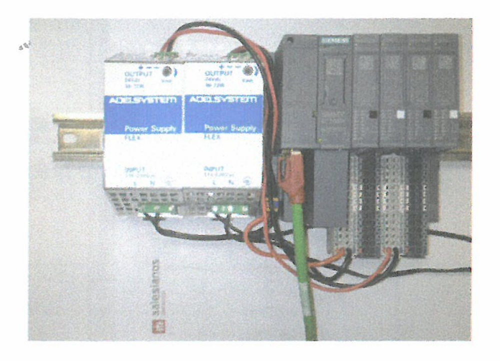


Figura 60

En la figura 61 se observa que se dispone de un módulo al final a la derecha que cierra los bloques. Ese módulo se debe colocar también en la configuración de TIA PORTAL. Se denomina *módulo servidor*. El ejercicio se va a realizar con esta disposición de módulos.

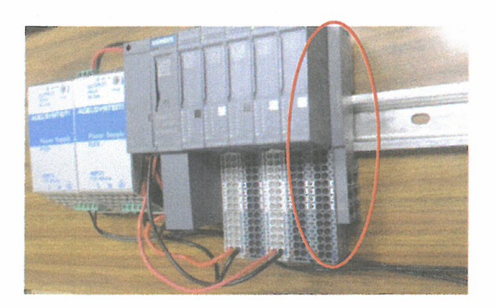


Figura 61

Configuración de la ET 200SP en TIA PORTAL

Volviendo a la configuración de TIA PORTAL, una vez situada la ET correcta en la *Vista de redes,* se pulsa dos veces sobre la ET y se pasa a la *Vista de dispositivos,* donde se colocan los diferentes módulos de entrada/salida. Ahora, el catálogo habrá cambiado y se indicarán los diferentes módulos que se pueden introducir en la ET. Se seleccionan del catálogo los adecuados. ¡Cuidado con las versiones del módulo Interfaz (IM-155) y también con los módulos de E/S! Se encuentran en la parte inferior del catálogo y aparecen una vez seleccionado el módulo (figura 62).

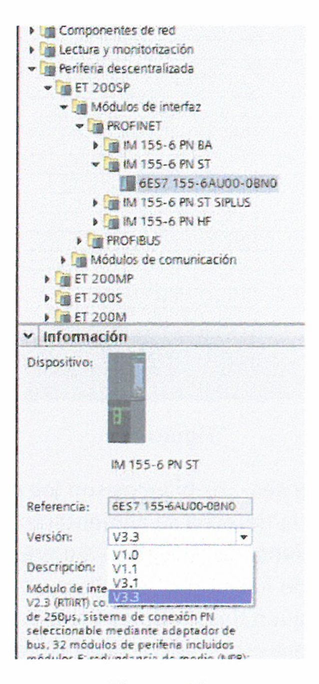


Figura 62

Cuando se colocan los diferentes módulos de E/S, los módulos de bloques se deben disponer de la forma que se aprecia en la figura 63.

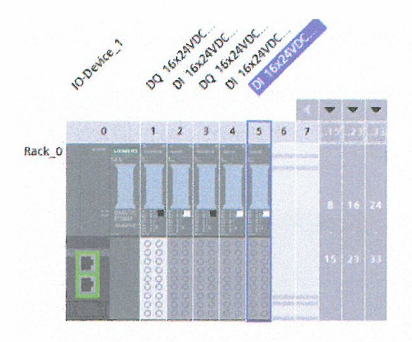


Figura 63

El primer bloque de bornes siempre debe ser blanco; luego, cambian todos a gris. Pero, en este caso, el bloque tercero también debe ser blanco (de alimentación). Para cambiarlo, se ha de ir a las propiedades del módulo, como se aprecia en la figura 64. Desde la pestaña *General/Grupo de potencial*, se selecciona «Permitir nuevo grupo de potencial» y el borneado cambiará a blanco.

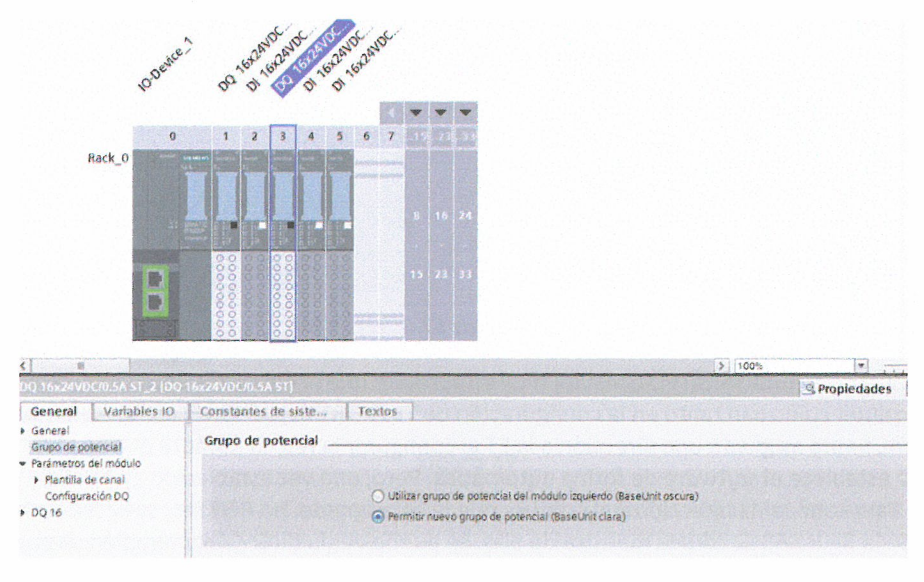


Figura 64

Seguidamente, se va a la *Vista de redes* y, desde allí, se crea la red entre la ET 200SP y el PLC 1516. Debe estar marcada la opción *Conectar en red*. Se realiza la conexión entre la ET y el PLC. Ahora se activa la opción *Conexiones* y se selecciona la *Conexión S7*. Para terminar, se vuelve a pulsar en *Conectar en red*. Quedará como en la figura 65.

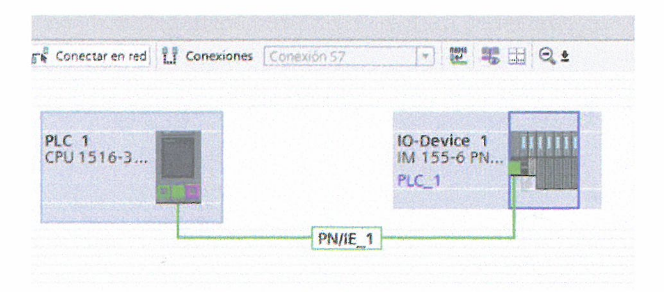


Figura 65

Para poder utilizar la ET y el PLC y crear los programas, es necesario saber las direcciones de las unidades de E/S, tanto de los módulos del PLC como de la ET. En la *Vista de dispositivos*, en el apartado *Vista general de dispositivos*, se observan esas direcciones, tal como indican las figuras 66 y 67. Estas direcciones no aparecerán si antes no se ha procedido a la conexión con el PLC.

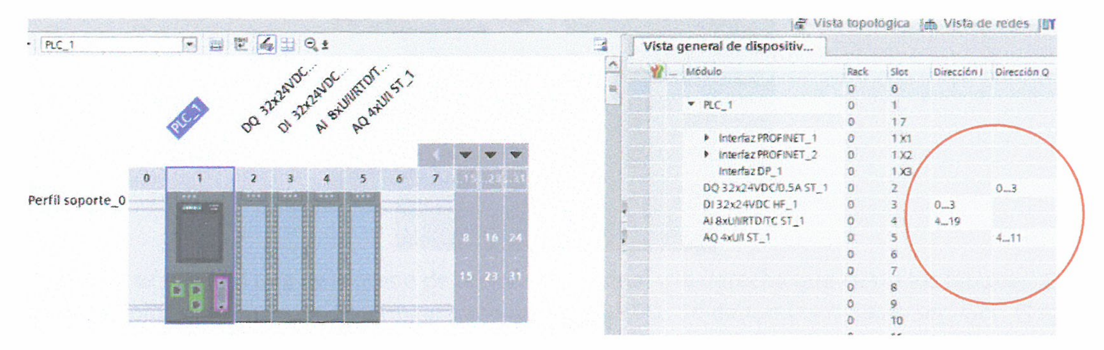


Figura 66

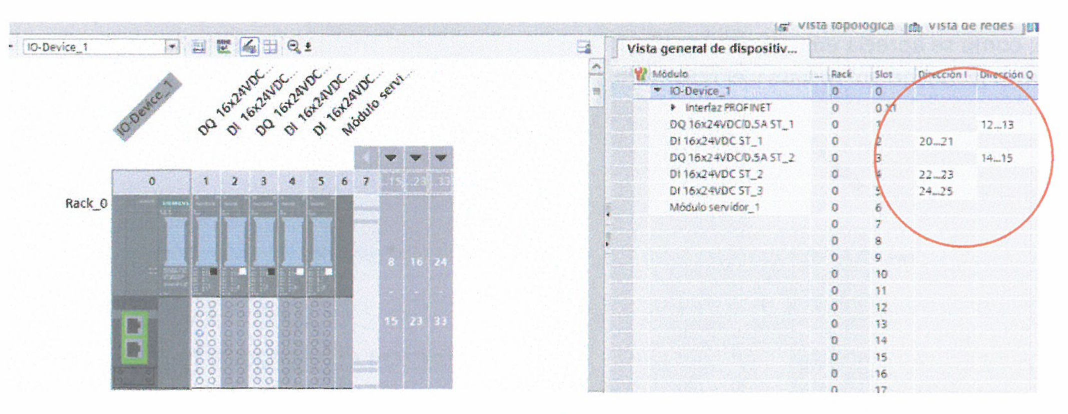


Figura 67

Antes de enviar la configuración al PLC, resulta muy importante que la dirección IP de la ET y el nombre de la estación en Profinet coincidan tanto en la configuración de TIA PORTAL (*OFFLINE*) como en la realidad (*ONLINE*). TIA PORTAL habrá asignado una dirección a la ET al configurar la red. El nombre que se pone a la ET en la red Profinet lo establece el *software* de forma automática. Pero, una vez establecido en TIA PORTAL, se debe cargar a la ET. Para conocer la dirección y el nombre que le ha asignado TIA PORTAL, se accede a las *Propiedades* de la ET. En la figura 68 se ofrece muestra de ello. Se puede ver la dirección y también se puede cambiar si se desea. Si lo que se quiere es cambiar el nombre de la ET, se ha de desmarcar la casilla *Generar automáticamente el nombre de la estación PROFINET*. Esto se encuentra un poco más abajo de la dirección IP.

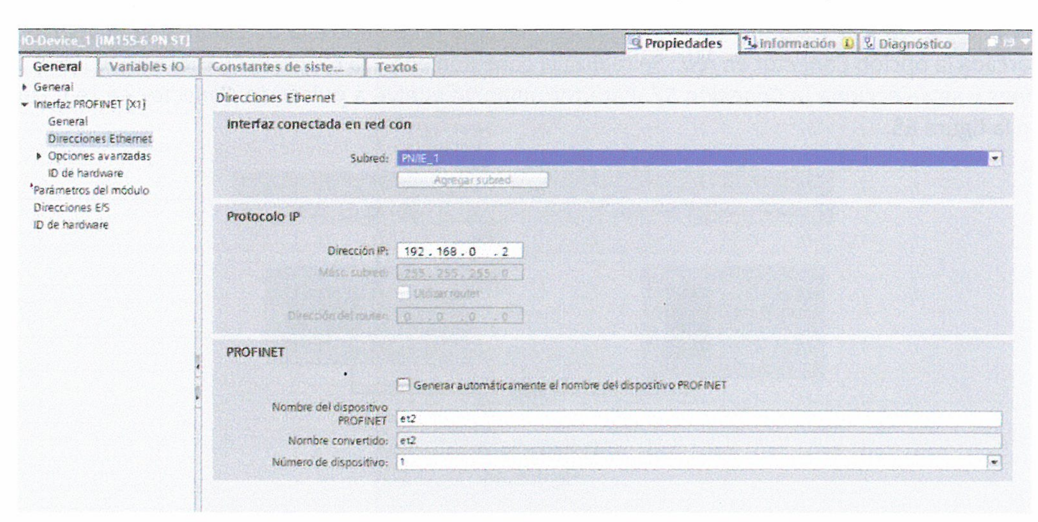


Figura 68

Unidad 4 · Ejercicios prácticos de Profinet

Para cargar la dirección y el nombre en la ET, se puede hacer desde *Accesos online,* ubicado en el árbol del proyecto, como se aprecia en la figura 69. Se pulsa en la interfaz con la que se comunica (PC Adapter o tarjeta de red del PC), en este caso la tarjeta de red del ordenador y, después, en *Actualizar dispositivos accesibles*; al poco tiempo, aparecerán los dispositivos accesibles desde esa tarjeta. Se pulsa en la ET y se acciona en *Online y diagnóstico*. Aparecerá la pantalla de *Accesos online* en la zona principal y, desde allí, se accede a las *Funciones,* donde se podrá poner la dirección y el nombre; debe ser el mismo que se ha puesto al configurar la ET y, si no coinciden, se copia de la configuración y se pega ahí. A continuación, se carga la dirección y el nombre en la ET.

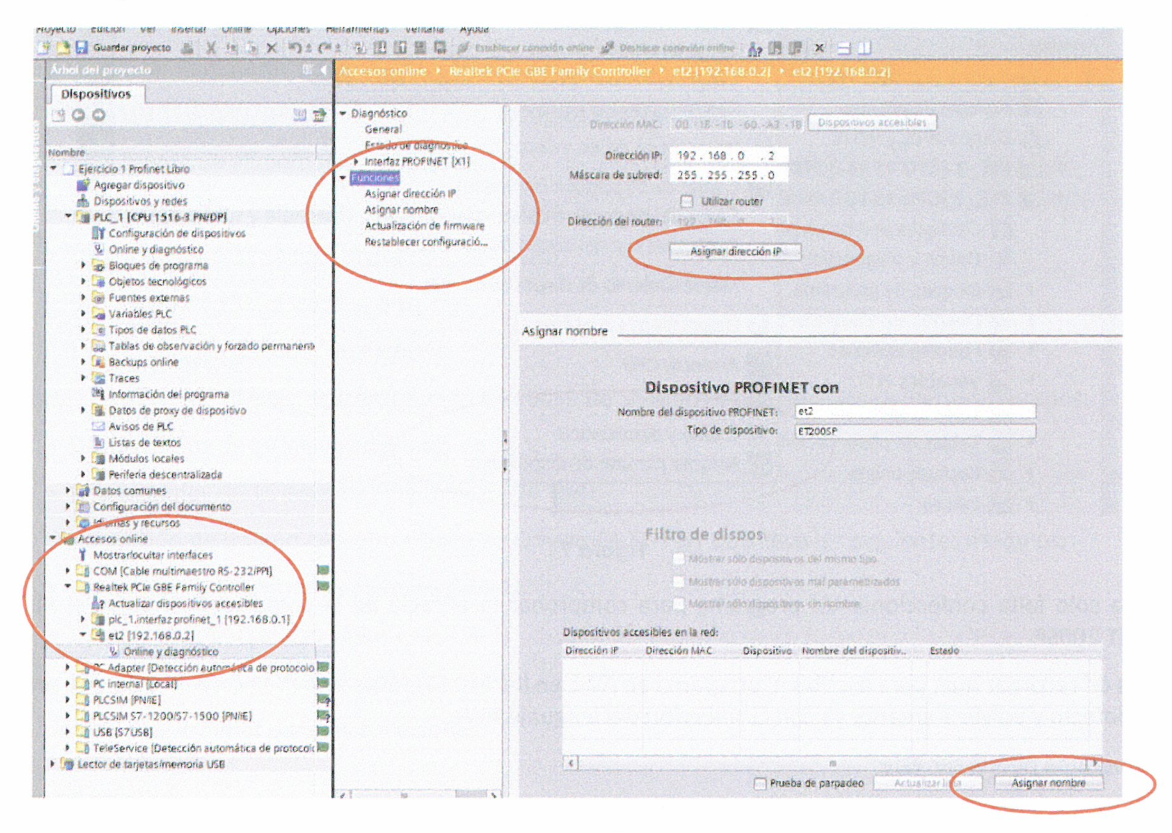


Figura 69

Para cambiar el nombre, también se puede acceder desde las diferentes vistas del *hardware* del proyecto (dispositivos, redes o topológica). En la figura 70 se aprecia el icono correspondiente al **cambio de nombre**.

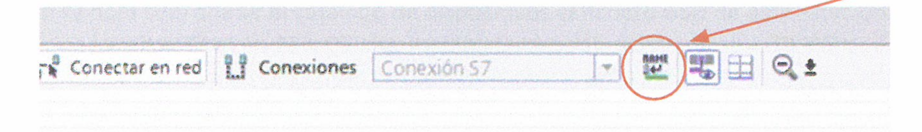


Figura 70

Otra opción para comprobar los dispositivos que se encuentran conectados en la red es desde los menús desplegables. En el menú *Online,* se hallan las opciones *Online y diagnóstico* y *Asignar nombre de dispositivo*. A través de la opción *Online* y diagnóstico, *se* accede a todos los dispositivos de la red; puede constituir una forma de verificar si toda la red es accesible y así detectar posibles errores (figura 71).

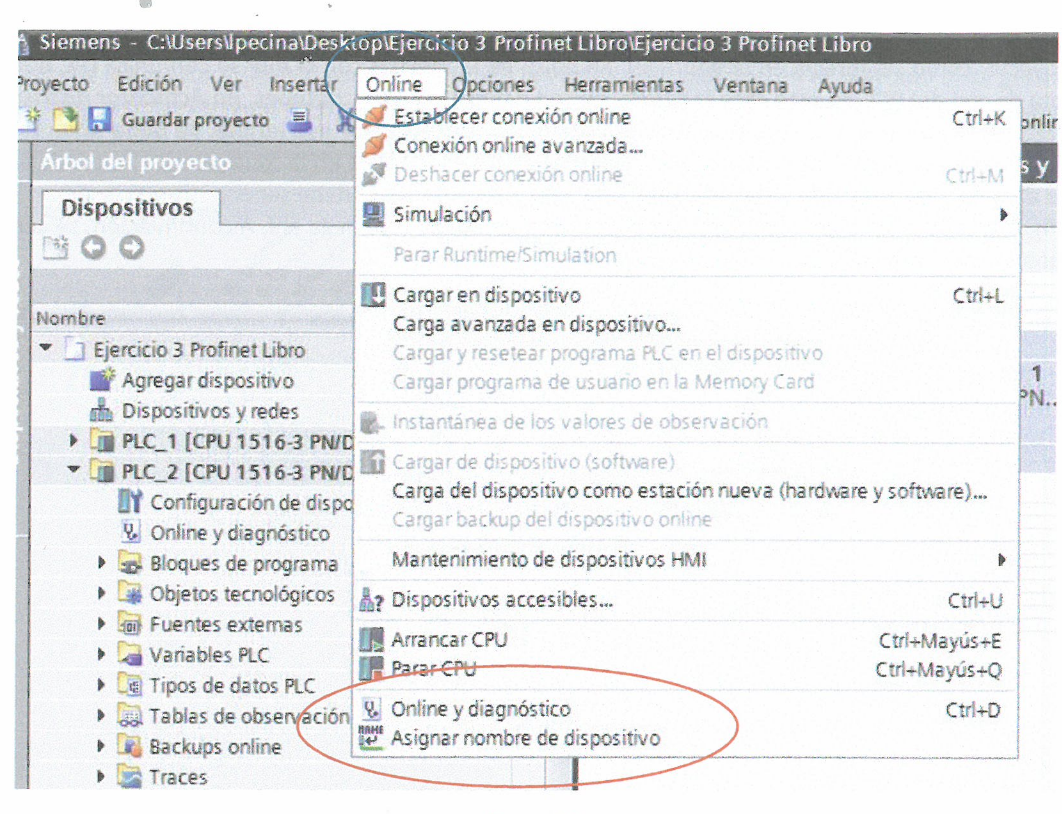


Figura 71

Ahora solo falta confeccionar un programa para comprobar la eficacia de la comunicación entre el PLC y la ET 200SP.

Se ha de recordar que, para escribir el programa en AWL en los PLC S7-1500, se debe borrar el OB1 que sale por defecto y volver a crearlo, ya con la selección del lenguaje AWL.

El programa puede ser este:

L EW0	Las direcciones de entradas 0 y 1 del PLC activan las
T AW12	direcciones de salidas 12-13 y 14-15 de la ET.
T AW14	
L EW20	Las direcciones de entrada 20 y 21 de la ET activan
T AW0	las direcciones de salida 0 y 1 del PLC.

4. Ejercicio de Profinet PLC + ET en sistema abierto

Enunciado

Al ejercicio anterior se le va a añadir otra estación de trabajo descentralizada de otro fabricante distinto. Esta estación nueva será del fabricante Omron.

Material necesario

Se va a utilizar el material enumerado a continuación:

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una estación de trabajo descentralizada ET 200SP de Siemens y que lleva los siguientes módulos:
   - Dos módulos de 16 salidas digitales de 24 V.
   - Dos módulos de 16 entradas digitales de 24 V.
3. Una estación de trabajo descentralizada de Omron GRT1-PNT que lleva los siguientes módulos:
   - Un módulo de 4 entradas digitales (ID4-1).
   - Un módulo de 4 salidas digitales (OD4-1).
   - Un módulo de 2 entradas analógicas (AD2).
4. Un cable de Profinet para las conexiones.

Realización

La diferencia de este ejercicio con respecto al anterior es la estación de Omron, que funciona conjuntamente con dispositivos *hardware* y *software* de otro fabricante; en este caso, Siemens. Pero se va a trabajar en una red abierta como es Profinet.

Lo primero que se hará será añadir al catálogo de dispositivos el fichero GSD de esta estación. Estos ficheros son los necesarios para poder trabajar en cualquier plataforma de un sistema abierto. Para ello, se buscará el fichero GSD en Internet. En la web de Omron se puede encontrar en la siguiente dirección: https://industrial.omron.es/es/products/smartslice#downloads.

Una vez encontrado, se debe descomprimir y guardar en el PC. Se abre el proyecto anterior de TIA PORTAL y se accede a la vista del proyecto. En la barra de menús, se accede a *Opciones* y, allí, a *Administrar archivos de descripción de dispositivos*. Aparecerá la ventana que se ve en la figura 72. Allí se busca la carpeta donde se encuentra el archivo GSD pulsando en los puntos suspensivos.

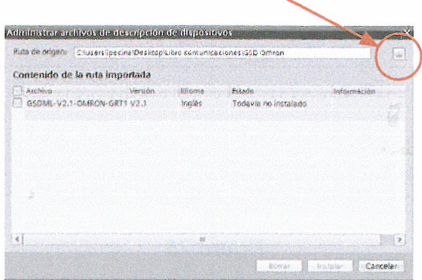


Figura 72

Una vez seleccionada la carpeta, aparecerá el GSD para instalar, tal como se aprecia en la figura 73.

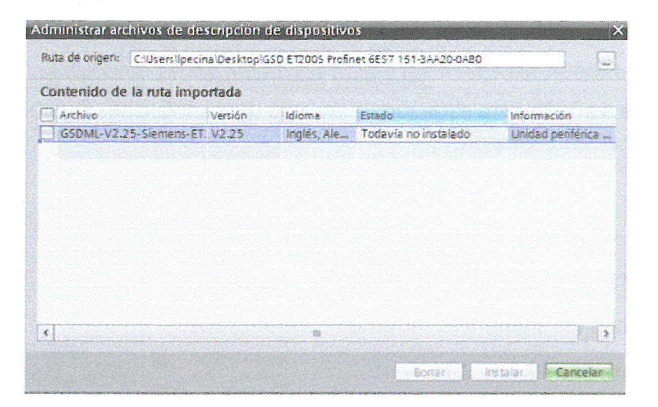


Figura 73

Se marca y se pulsa *Instalar*. Tras hacerlo, se busca en el catálogo desde *Vista de redes* o desde *Vista de dispositivos.* En la figura 74 se puede ver la disposición de la estación de Omron en el catálogo.

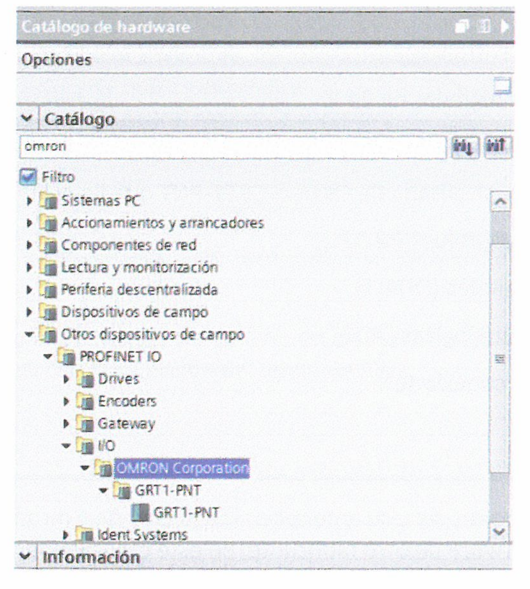


Figura 74

Después de haberlo ubicado, se emplaza en el proyecto. Ahora, desde *Vista de dispositivos*, se añaden los diferentes módulos de los que dispone la estación descentralizada. En este caso, se deben colocar en las celdas que aparecen a la derecha en *Vista general de dispositivos*. El catálogo se compone de los diferentes módulos de la estación Omron. En la figura 75 se describe cómo debe llevarse a cabo.

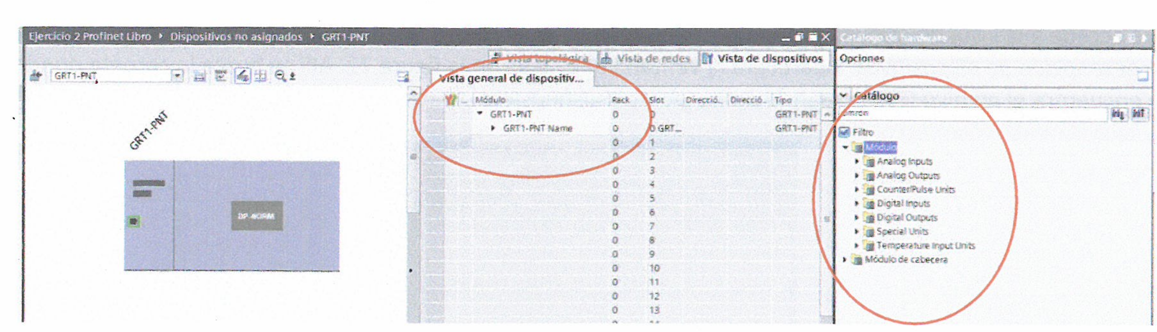


Figura 75

Se colocan los diferentes módulos en las filas de *Vista general de dispositivos*. No hay que olvidar el módulo final, que aquí se denomina «fin de módulo» (END-M) y que se sitúa en la carpeta *Special Units*. El módulo

de entradas analógicas se emplaza en *Default*. En todos los módulos existen dos opciones: una indicada con un asterisco y otra sin asterisco; se deben introducir todos de la misma forma, con asterisco o sin él. En la figura 76 se representa la estación con todos sus módulos.

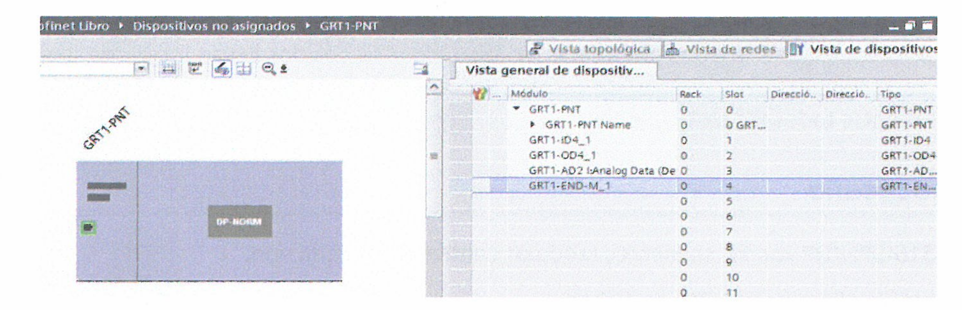


Figura 76

Desde *Vista de redes,* se realiza la conexión de la estación a la red, tal como se procedió con la estación de Siemens. La dirección se la pondrá el propio TIA PORTAL al realizar la conexión; lo mismo pasará con el nombre de la estación. Después se deberá enviar dicha dirección y nombre a la estación siguiendo el procedimiento anterior desde *Accesos online*. Aunque, en este caso, solo se dispone de un PLC, se debe asignar ese PLC a la estación. Para ello se pulsa sobre *No asignado* y se selecciona la opción que aparece, correspondiente al único PLC que hay en el proyecto. En la figura 77 se indica el procedimiento.

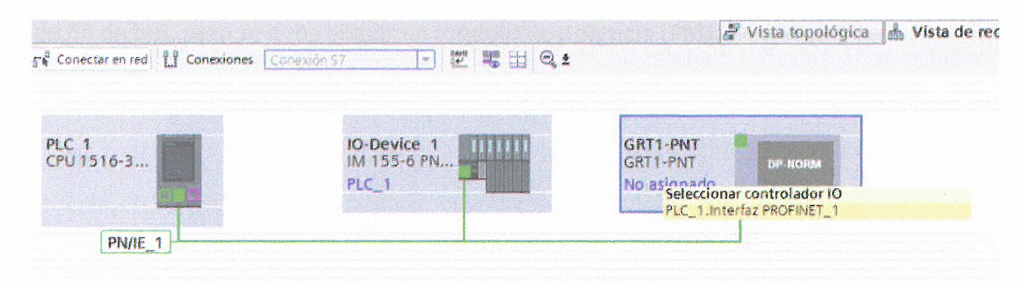


Figura 77

Ahora solo queda diseñar un programa donde se añada las entradas y salidas de la nueva estación de Omron.

Programa con símbolos	Direcciones	Comentarios
L E_PLC T S_ET 200_1 T S_ET 200_2 T S_Omron	EW0 AW12 AW14 AB16	Las direcciones de entradas 0 y 1 del PLC activan las direcciones de salidas 12-13, la 14-15 de la ET 200 y la 16 de la estación de Omron.
L E_ET 200 T S_PLC T S_Omron	EW20 AW0 AB17	Las direcciones de entrada 20 y 21 de la ET activan las direcciones de salida 0 y 1 del PLC y la 17 de la estación de Omron.
L E_Omron T S_PLC T S_ET 200	EB32 AW0 AB12	Las direcciones de entrada 32 de la estación de Omron activan las salidas 0-1 del PLC y las direcciones 12 de la ET 200.

5. Ejercicio de Profinet de dos PLC y dos ET

Enunciado

Se va a realizar la conexión entre dos autómatas S7 1516-3PN/DP. Los dos llevan integrada una interfaz de Profinet I/O. Se le asignará una estación de trabajo descentralizada a cada uno de ellos. Deberá poderse intercalar datos entre todos los participantes.

Material necesario

1. Dos CPU 1516-3PN/DP con los siguientes módulos centralizados cada uno:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una estación de trabajo descentralizada ET 200SP de Siemens y que lleva los siguientes módulos:
   - Dos módulos de 16 salidas digitales de 24 V.
   - Dos módulos de 16 entradas digitales de 24 V.
3. Una estación de trabajo descentralizada ET 200S de Siemens y que lleva los siguientes módulos:
   - Cinco módulos de salidas digitales de 4 bits.
   - Cinco módulos de entradas digitales de 4 bits.
4. Un cable de Profinet para las conexiones.

Realización

Se trata de realizar una comunicación entre dos PLC S7 1516- 3PN/DP y dos ET: una ET 200SP y otra ET 200S. Del ejercicio anterior se mantiene el PLC y la ET 200SP.

Para realizar la comunicación entre los dos PLC, se necesita confeccionar una conexión/enlace entre ambos autómatas. En este caso se producirá un enlace de tipo S7. Estos enlaces se llevan a cabo entre dispositivos Siemens. Para ello se utilizan los bloques de comunicación PUT y GET.

Se comienza configurando los dos autómatas. Uno de ellos será el interlocutor y el otro, el local. A cada uno se le debe poner una dirección IP. Después, se añade la ET 200S y se incluyen los módulos de entrada y salida que lleve. En este caso, incorpora 5 módulos de entradas digitales de 4 bits cada uno y otros 5 módulos de salidas digitales de 4 bits cada uno. La ET 200SP se asigna al PLC1, como en el ejercicio anterior. La nueva ET 200S se le asigna al nuevo PLC2. Cada dispositivo debe presentar una dirección distinta pero dentro del grupo de la red. Las ET han de mostrar cada una un nombre diferente y cargarlo a cada una de ellas. Para ingresar direcciones y nombres en las ET, se debe seguir el mismo criterio que para los ejercicios anteriores. Se elabora la red entre los diferentes dispositivos. Primero, se unen los PLC y, posteriormente, se junta cada ET con el PLC que le corresponda; de este modo se le asigna a cada ET su autómata.

En la figura 78 se indica cómo quedarán los dos autómatas al lado de las dos ET.

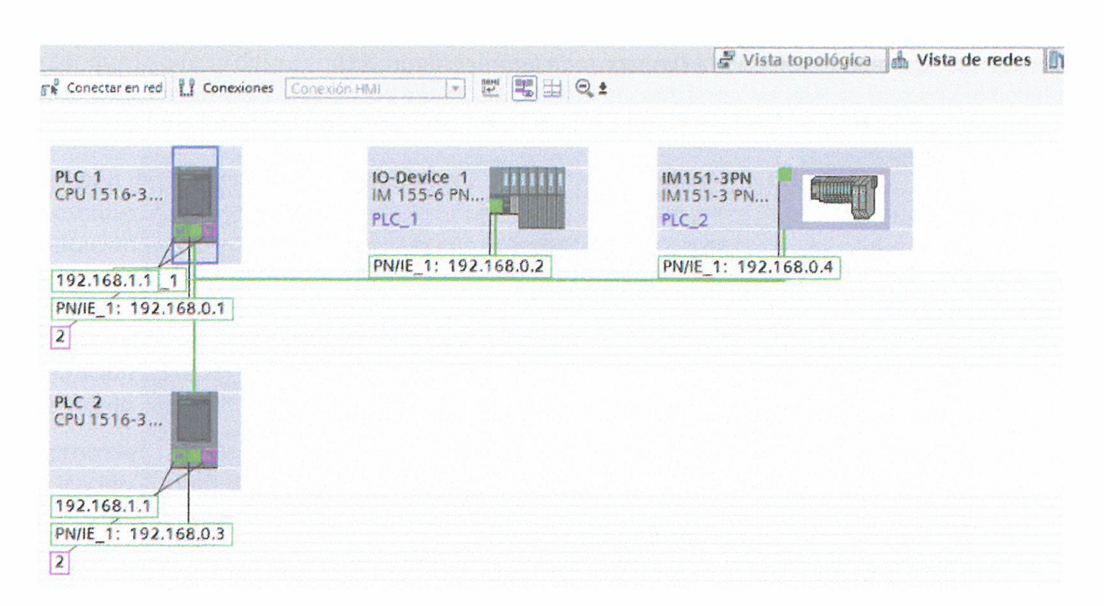


Figura 78

Configuración de la ET 200S en TIA PORTAL

En la ET 200S no se seleccionan grupos de potencial como en la ET 200SP. Tampoco hay que configurar el terminal de fin de bus, pero se debe añadir un **módulo de potencia (PM)**, tal como se aprecia en la figura 79.

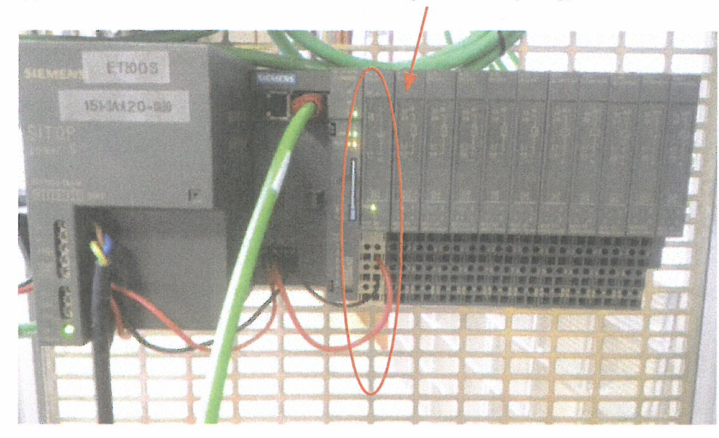


Figura 79

Puede suceder que se disponga de una ET que no exista en el catálogo, como ocurre con la ET de la figura 75, cuya referencia 151-3AA20-0AB0 no existía en el catálogo, aun siendo de Siemens. Para este caso, se procede de la misma forma que se ha hecho para el de la estación de Omron. Lo primero que se debe hacer es actualizar el catálogo. Para ello, se buscan los *Paquetes de soporte* para TIA PORTAL (HSP) con el fin de actualizar el catálogo. Eso se hace desde el propio TIA PORTAL, en *Opciones* y, después, en *Support Packages*. En la figura 80 se detalla el camino.

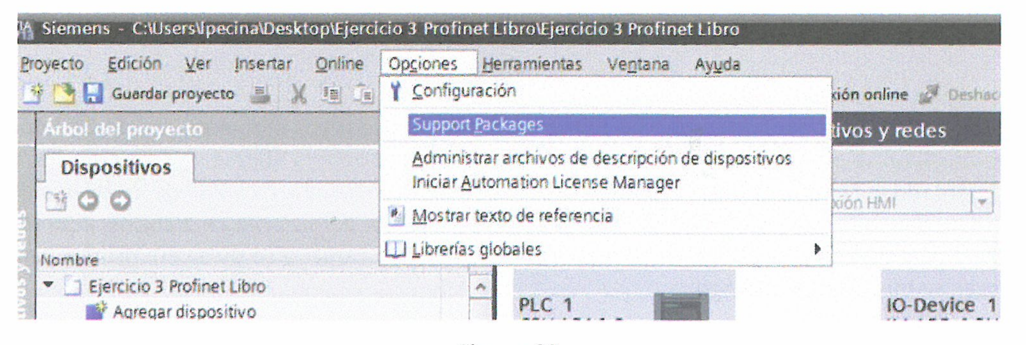


Figura 80

En la ventana que aparece, se selecciona *Descargar de Internet* (figura 81).

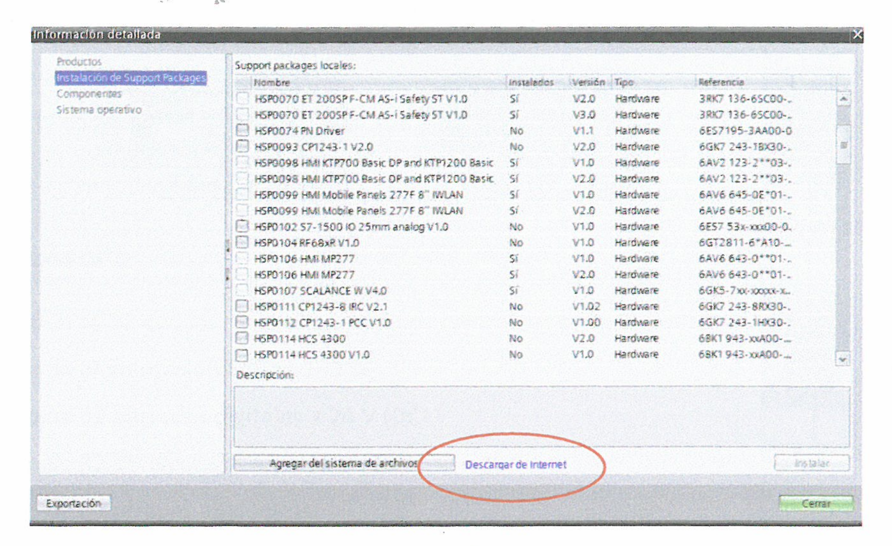


Figura 81

Si se dispone de conexión a Internet, se conectará con la web de Siemens; concretamente, con el lugar donde se sitúa el HSP más reciente y para las diversas versiones de TIA PORTAL. Hay que fijarse en la fecha del artículo (figura 82).

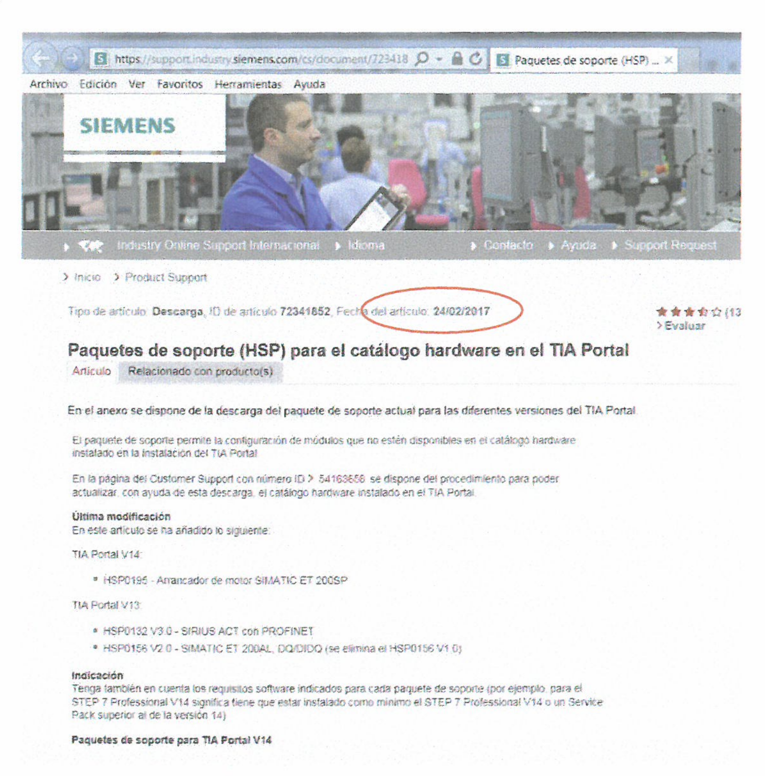


Figura 82

Cuando se encuentre la opción adecuada a nuestra versión se descarga, momento en el que le pedirá la identificación. Si no se posee cuenta, no se podrá bajar, por lo que deberá abrirse una; es gratuita. Una vez bajado y guardado el fichero, se descomprime. Ahora se cuenta con los diversos HSP de los dispositivos que se desean actualizar. Se retorna a TIA PORTAL y se accede a *Agregar del sistema de archivos*; aquí se buscan los HSP descargados y descomprimidos. Se marca el que se desee cargar y se pulsa en *Instalar* (figura 81). Para finalizar, se busca en el catálogo.

Pudiera ser que lo que se desea encontrar no se halle en los paquetes HSP. En estos, suele haber dispositivos de reciente creación. En los casos en los que no se pudiera encontrar en los HSP, se buscaría el fichero GSD de la estación y se procedería de igual forma a lo visto para la estación de Omron.

Configuración del PLC *local* y del *interlocutor*

Una vez establecida la red, se debe configurar qué PLC hace de *local* y cuál de *interlocutor*. Para establecer la comunicación entre los dos PLC, se realiza una conexión de tipo S7 y, para completar el proceso, se utilizan unos bloques de *software* para la comunicación. Estos bloques son los GET y PUT que se encuentran en la librería de comunicaciones. Se colocan solo en OB1 del PLC que hace de *local*. Se debe recordar que, en Profinet, no resulta correcto hablar de maestro y esclavo, como en algunos sitios se ve. La particularidad de la comunicación Ethernet, en la que se basa Profinet, impide denominarlos de ese modo, ya que un maestro es el que «*toma el poder*» y manda en el bus y, en Ethernet, todos son «*dueños*» del bus.

Se debe proceder a colocar los bloques GET y PUT en el OB1 del *local* y se añaden las variables necesarias para la comunicación. En la figura 83 se muestra dónde se encuentran. Para colocarlos en el OB1, el mejor método consiste en arrastrar.

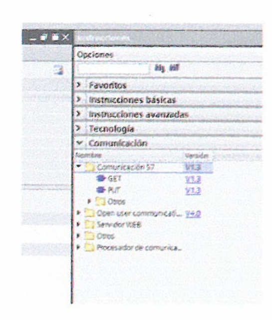


Figura 83

Una vez que se han colocado los dos bloques, se procede a añadir las diferentes variables de acuerdo con los tipos que se necesitan en cada parámetro. En la descripción de cada bloque que se realiza a continuación, se explica cada parámetro.

Antes de pasar a ver esa descripción, se termina de configurar los PLC. Tras disponer los dos bloques en el OB1, se **toca con el ratón en GET o PUT** y, en el área de propiedades, se **añade el interlocutor (si no aparece)** y se fija en la **dirección de la conexión (ID).** En la figura 84 se observa cómo queda.

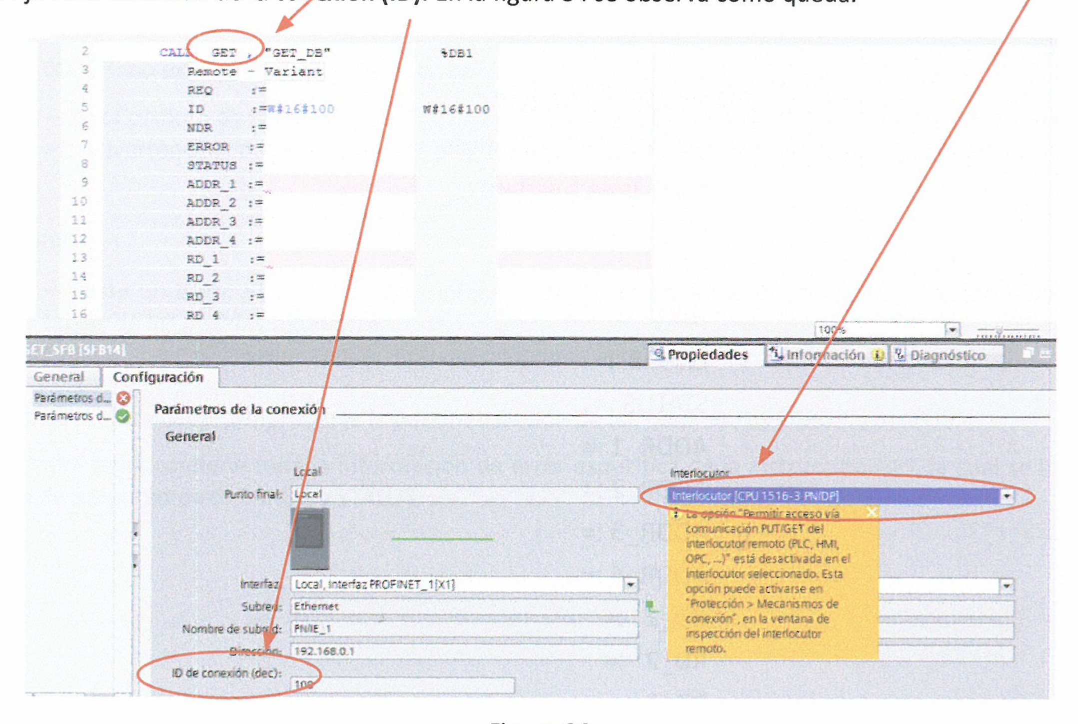


Figura 84

Si aparece una nota en un cuadrado anaranjado, como se ve en la figura 84, hay que marcar la opción de «*Permitir acceso vía comunicación PUT/GET del interlocutor remoto...*» en el PLC interlocutor. Esto se efectúa en las propiedades de la CPU en el apartado de *Protección*, como se muestra en la figura 85. Una vez realizado ese cambio, se procede a completar los parámetros necesarios en los bloques GET y PUT. Se ha de destacar que esto se debe hacer solo en el PLC interlocutor y no en el local.

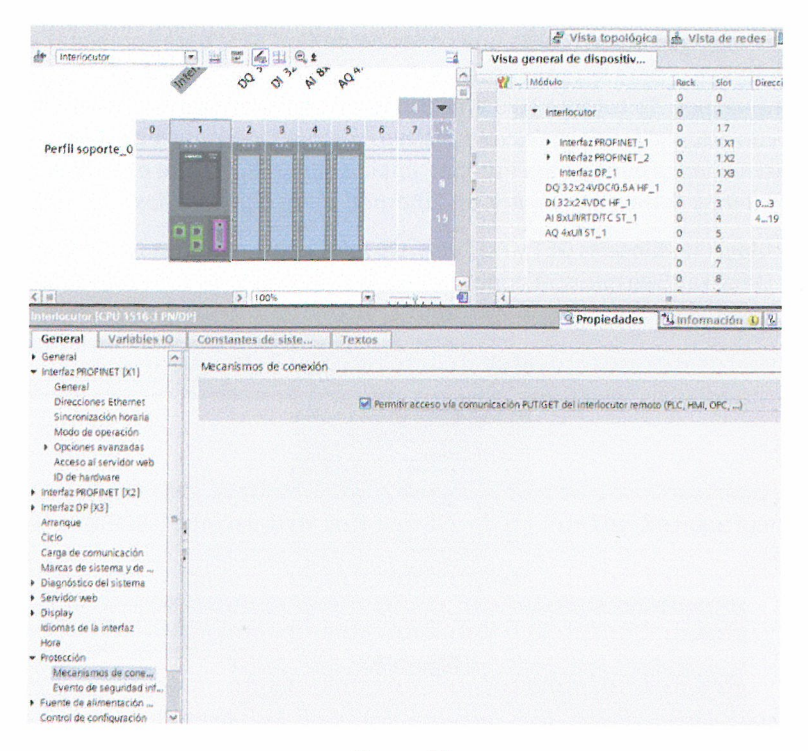


Figura 85

Descripción de los bloques GET y PUT

El bloque GET lee datos de una CPU remota y el bloque PUT escribe datos en una CPU remota. Los parámetros de estos bloques son los enumerados a continuación:

Para GET:

```
CALL GET , "GET_DB"
  remote_type:=Remote
  local_type:=Variant
  REQ   :=
  ID    :=
  NDR   :=
  ERROR :=
  STATUS :=
  ADDR_1 :=
  ADDR_2 :=
  ADDR_3 :=
  ADDR_4 :=
  RD_1  :=
  RD_2  :=
  RD_3  :=
  RD_4  :=
```


Para PUT:

```
CALL PUT , "PUT_DB"
  remote_type:=Remote
  local_type:=Variant
  REQ   :=
  ID    :=
  DONE  :=
  ERROR :=
  STATUS :=
  ADDR_1 :=
  ADDR_2 :=
  ADDR_3 :=
  ADDR_4 :=
  SD_1  :=
  SD_2  :=
  SD_3  :=
  SD_4  :=
```

Estos dos bloques se ejecutan con la activación de la señal colocada en el parámetro REQ. Para que sea de forma secuencial, se coloca un bit de marca de ciclo; por ejemplo, el de 10 Hz. En ID se debe insertar la dirección del enlace que se crea de forma automática al realizar la conexión. Se ha de transcribir en formato hexadecimal. Esto puede aparecer al poner los bloques y configurar el enlace.

El bit de ERROR y de STATUS es igual para los dos bloques. Estos parámetros muestran la siguiente indicación:

- ERROR=0

  STATUS tiene el valor:

  0000H: no existe ni advertencia ni error.

  —Valor distinto de 0000H: constituye una advertencia. STATUS devuelve información detallada sobre esa advertencia. En las tablas de las páginas siguientes se observan las diferentes advertencias.

- ERROR=1

  —Existe un error. STATUS devuelve información detallada sobre el tipo de error, como indican las tablas de las páginas siguientes.

En dichas tablas puede comprobarse la parametrización de los parámetros ERROR y STATUS.

La primera tabla contiene toda la información de error específica de la instrucción GET, la cual se indica a través de los parámetros ERROR y STATUS.

ERROR	STATUS (decimal)	Explicación
0	11	Advertencia: la nueva petición no posee efecto, ya que la anterior todavía no ha finalizado.
0	25	La comunicación se ha iniciado. La petición se encuentra en proceso.
1	1	Problemas de comunicación: • Descripción de la conexión no cargada (local o remota) • Conexión interrumpida (p. ej., cable, CPU desconectada, CP en STOP) • Conexión con el interlocutor todavía no establecida
1	2	• Acuse negativo del interlocutor. La función no puede ejecutarse. • La respuesta de la estación remota supera la longitud de datos de usuario máxima (consulte: *Parámetros comunes de las instrucciones para la comunicación S7*). • La protección contra acceso se halla activada en la CPU interlocutora. Desactive la protección contra acceso en los ajustes de la CPU.
1	4	Error en los punteros de almacenamiento de datos RD_i: • Los tipos de datos de los parámetros RD_i y ADDR_i no se muestran compatibles entre sí. • La longitud del área RD_i resulta menor que la longitud de los datos que se deben leer de los parámetros ADDR_i.
1	8	Error de acceso en la CPU interlocutora.
1	10	El acceso a la memoria de usuario local no es posible (p. ej., acceso a un DB borrado).
1	20	Se ha excedido el número máximo de peticiones paralelas. La petición se realizará en una clase de baja prioridad (primera llamada).

Extraído del manual/ayuda de TIA PORTAL (Siemens).

La siguiente tabla contiene toda la información de error específica de la instrucción PUT, que puede indicarse a través de los parámetros ERROR y STATUS.

ERROR	STATUS (decimal)	Explicación
0	11	Advertencia: la nueva petición no posee efecto, ya que la anterior todavía no ha finalizado.
0	25	La comunicación se ha iniciado. La petición se encuentra en proceso.
1	1	Problemas de comunicación: • Descripción de la conexión no cargada (local o remoto). • Conexión interrumpida (p. ej., cable, CPU desconectada, CP en STOP). • Conexión con el interlocutor todavía no establecida.
1	2	Acuse negativo de la CPU interlocutora. La función no puede ejecutarse. El acceso en la CPU interlocutora no se ha permitido. Active el acceso en los ajustes de la CPU.
1	4	Error en los punteros de almacenamiento de datos: • Los tipos de datos de los parámetros SD_i y ADDR_i no resultan compatibles entre sí. • La longitud del área SD_i se muestra mayor que la longitud de los datos que se deben escribir de los parámetros ADDR_i. • No se puede acceder a SD_i. • Se ha excedido el volumen máximo de datos de usuario. • El número de parámetros SD_i y ADDR_i no coinciden.
1	8	Error de acceso en la CPU interlocutora (p. ej., no se ha cargado el DB o está protegido contra escritura).
1	10	El acceso a la memoria de usuario local no es posible (p. ej., acceso a un DB borrado).
1	20	Se ha excedido el número máximo de peticiones paralelas. La petición se realizará en una clase de baja prioridad (primera llamada).

Extraído del manual/ayuda de TIA PORTAL (Siemens).

Parámetros propios de GET

✓ Parámetro NDR: constituye un bit de estado con el siguiente significado:

- 0: la petición aún no se ha iniciado o todavía se halla en curso.
- 1: la petición ha finalizado correctamente.

✓ Parámetros de ADDR_1 a ADDR_4: un puntero que señala las áreas de la CPU interlocutora que se deben leer. No se necesita establecer las 4 áreas.

Ejemplo

Utilizando marcas P#M100.0 *Byte* 5: su significado es que se apunta al bit M100.0 del PLC interlocutor y se utilizan los 5 *bytes* siguientes; es decir, del *byte* 5 al *byte* 10.

Utilizando DB: P#DB10.DBX5.0 *Byte* 10: el significado es el indicado anteriormente, pero apunta a una posición del DB.

Parámetros de RD_1 a RD_4: un puntero que señala las áreas de la CPU propia (local), donde se depositarán los datos leídos de la CPU interlocutora. También debe indicarse en formato puntero con el mismo significado indicado anteriormente. No se necesita establecer las 4 áreas.

Parámetros propios de PUT

✓ Parámetro DONE: constituye un bit de estado que significa lo siguiente:

- 0: la petición aún no se ha iniciado o todavía se está ejecutando.
- 1: la petición ha finalizado sin errores.

✓ Parámetros de ADDR_1 a ADDR_4: un puntero que señala las áreas de la CPU interlocutora en las que debe escribirse. No se necesita establecer las 4 áreas.

✓ Parámetros de SD_1 a SD_4: un puntero que señala las áreas de la CPU propia (local) que contienen los datos que deben transmitirse. No se necesita establecer las 4 áreas.

Programa del PLC local y del PLC interlocutor

OB1 del PLC local

Una vez explicados los dos bloques que se deben utilizar en un enlace del tipo S7, se muestra un ejemplo para poder comunicar los dos PLC y la ET 200.

Los bloques GET y PUT se deben colocar en el PLC local y, en este caso, los parámetros elegidos se indican en el siguiente programa.

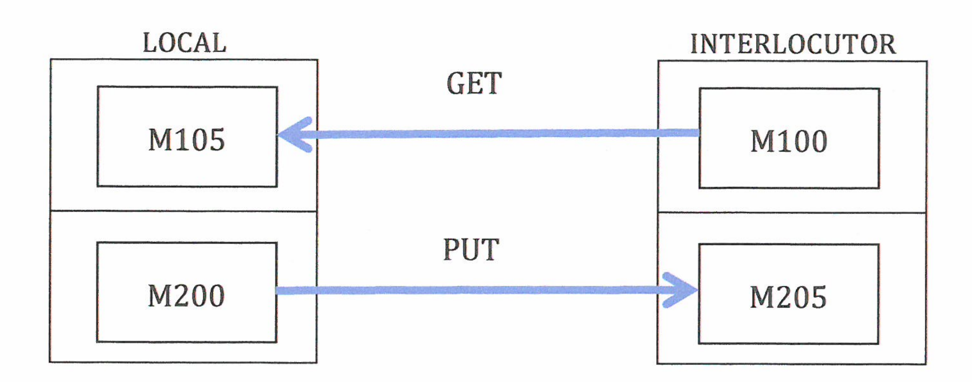

```

    CALL  GET , "GET_DB"                          %DB1
        Remote - Variant
        REQ     :="Clock_10Hz"                    %M20.0
        ID      :=W#16#100                        W#16#100
        NDR     :="Tag_2"                         %M1.0
        ERROR   :="Tag_3"                         %M1.1
        STATUS  :="Tag_5"                         %MW2
        ADDR_1  :=P#M100.0 BYTE 4                 P#M100.0...
        ADDR_2  :=
        ADDR_3  :=
        ADDR_4  :=
        RD_1    :=P#M105.0 BYTE 4                 P#M105.0...
        RD_2    :=
        RD_3    :=
        RD_4    :=


    CALL  PUT , "PUT_DB"                          %DB2
        Remote - Variant
        REQ     :="Clock_10Hz"                    %M20.0
        ID      :=W#16#100                        W#16#100
        DONE    :="Tag_7"                         %M4.0
        ERROR   :="Tag_9"                         %M4.1
        STATUS  :="Tag_10"                        %MW5
        ADDR_1  :=P#M205.0 BYTE 4                 P#M205.0...
        ADDR_2  :=
        ADDR_3  :=
        ADDR_4  :=
        SD_1    :=P#M200.0 BYTE 4                 P#M200.0...
        SD_2    :=
        SD_3    :=
        SD_4    :=

```

Según las direcciones de lectura/escritura dispuestas en el ejercicio, el intercambio de datos queda de este modo:



Además de los bloques GET y PUT, en el OB1 del PLC local se han de colocar las instrucciones para comunicarse con el PLC interlocutor y con la ET 200 que, en este caso, pertenece al PLC local y, por lo tanto, se sitúa en su dominio de direcciones de E/S.

Estas órdenes son:

```
// Una entrada del PLC Local envía a la zona de salidas
// del PLC Local (hacia el Interlocutor).
      U      "E_PLC_Local_1.7"                    %E1.7
      =      "Memoria_Salida_Local_200.0"         %M200.0

// Un dato del PLC Interlocutor activa una salida
// del PLC Local.
      U      "Memoria_Entrada_Local_105.0"        %M105.0
      =      "S_PLC _Local"                       %A0.0

//Una entrada del PLC Local acitva una salida de su ET 200
// y a la vez se envía a la zona de salidas del Local.
      U      "E_PLC_Local_0.0"                    %E0.0
      =      "S_ET_Local_12.0"                    %A12.0
      =      "Memoria_Salida_Local_200.1"         %M200.1

// Un dato del PLC Interlocutor activa una salida de
// la ET Local.
      U      "Memoria_Entrada_Local_105.1"        %M105.1
      =      "S_ET_Local_12.1"                    %A12.1
```

OB1 del PLC interlocutor

El OB1 del interlocutor no debe presentar bloques de comunicación; solo constará de las órdenes que intercambien los datos entre las CPU y ET.

El programa es el siguiente:

```
// Un dato del PLC Local activa una
// salida del PLC Interlocutor.
      U      "Memoria_Entrada_Interl_205.0"       %M205.0
      =      "S_PLC _Interl_8.0"                  %A8.0
// Una entrada del PLC Interlocutor activa una
//salida de la ET del Interlocutor y y a la vez
//se envía a la zona de salidas del Interlocutor.
      U      "E_PLC_Interl_1.7"                   %E1.7
      =      "Memoria_Salida_Interl_100.0"        %M100.0
      =      "S_ET_Interl_12.0"                   %A12.0
// Una entrada del PLC Local activa una salida de
// la ET Local.
      U      "Memoria_Entrada_Interl_205.1"       %M205.1
      =      "S_ET_Interl_12.1"                   %A12.1
// Una entrada del PLC interlocutor envía a la zona
//de salidas del Interlocutor.
      U       E_interlocutor_1.6                  %E1.6
      =      "Memoria_salida_Interl_100.1"        %M100.1
```

Ejercicio propuesto

En el ejemplo anterior, faltan las órdenes que hacen que las ET, tanto la perteneciente al PLC local como al PLC interlocutor, activen salidas de los PLC y ET.

Se propone completar el ejercicio añadiendo las órdenes necesarias para que una entrada de la ET perteneciente al PLC local active:

- Una salida propia de la ET 200.
- Una salida de la ET del interlocutor.
- Una salida de su PL.
- Una salida del PLC interlocutor.

Igualmente, se pretende que una entrada de la ET perteneciente al PLC interlocutor active:

- Una salida propia de la ET 200.
- Una salida de la ET del local.
- Una salida de su PLC.
- Una salida del PLC local.

6. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada

Enunciado

En el siguiente ejercicio se van a comunicar dos PLC, un S7-1500 y un S7-300; concretamente, el PLC S7 1516 3PN/DP y el S7-315 2PN/DP.

El S7-1500 se configurará y programará en TIA PORTAL y el S7-300 utilizará el *software* clásico de STEP 7 V5.5. Este caso puede darse cuando el PLC no resulte compatible con TIA PORTAL.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una CPU 315 2PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación SITOP 5 A.
   - Un módulo de 16 entradas digitales a 24 V (DC).
   - Un módulo de 16 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas y 2 salidas analógicas.
3. Un cable de Profinet para las conexiones.

Realización

Para realizar este ejercicio, debe configurarse el PLC S7-315 2PN/DP con el *software* del STEP 7 V5.5 o similar. No se ha de hacer nada especial que no sea colocar sobre el bastidor los diferentes módulos de los que disponga el autómata. Tenga presente, no obstante, que hay que introducir una dirección IP dentro del mismo grupo que se le vaya a poner al PLC S7-1516; por ejemplo, en este caso, se pondrá la dirección 192.168.0.2 al PLC S7-315, y al PLC S7-1516, la dirección IP 192.168.0.1.

Para configurar el PLC S7-1516 3PN/DP se hará como en los ejercicios anteriores, disponiendo la dirección IP 192.168.0.2.

Una vez configurado dicho PLC en TIA PORTAL, quedará como se indica en la figura 86.

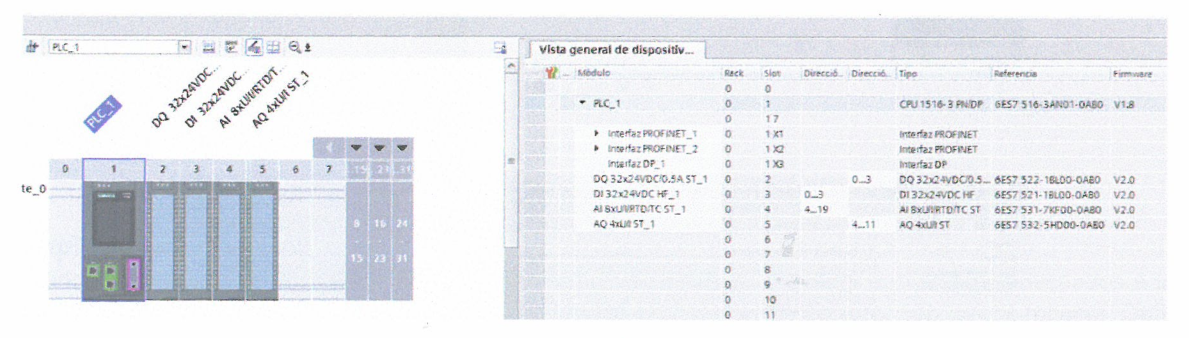


Figura 86

En redes se debe sacar una red de Profinet con una conexión de tipo S7. Se refleja en la figura 87.

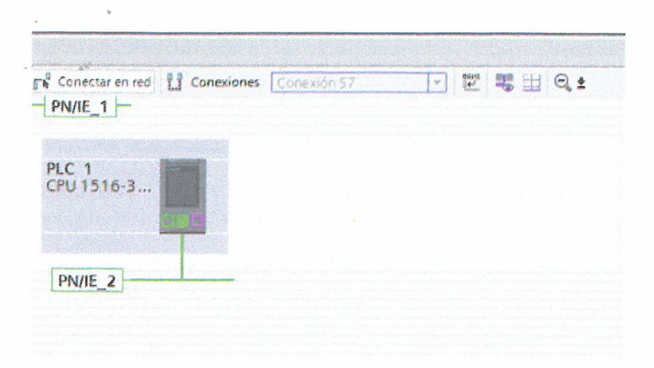


Figura 87

En el OB1, se deben colocar los bloques GET y PUT. Pulsando con el ratón en GET o PUT y observando las propiedades en el área inferior, se debe rellenar de la forma como se expone en la figura 88.

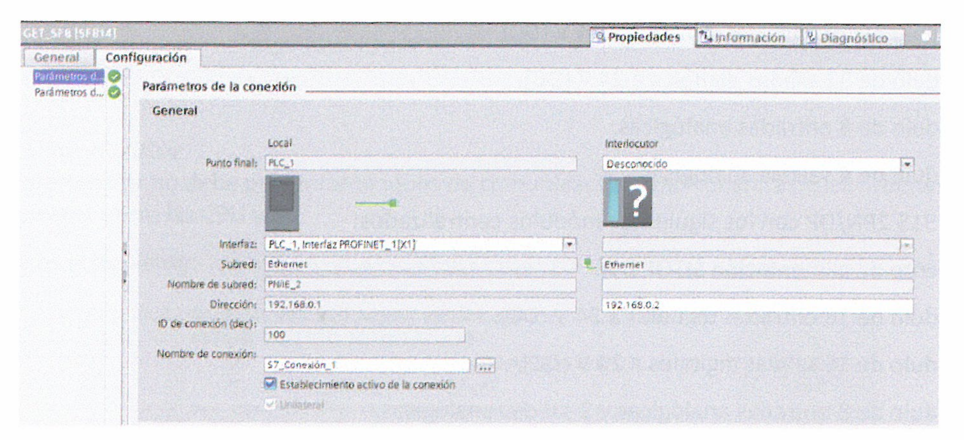


Figura 88

Se observa que, como interlocutor, la estación es «Desconocida» porque es un S7-315 que se ha configurado con otro *software*. No obstante, se ha de colocar la dirección real de ese PLC «desconocido», 192.168.0.2, en este caso. También debe marcarse que el PLC local será el que establezca la conexión.

Por último, se comprobará que el TSAP (puede leerse más sobre TSAP al final de este ejercicio) esté como se indica en la figura 89. Para acceder a ese lugar se debe ir a *Vista de redes* y en la parte derecha, se accede a la solapa *Conexiones*. Presionando sobre la conexión *S7_Conexión_1,* se observan las *Propiedades*. En el apartado siguiente se amplía el conocimiento sobre el TSAP.

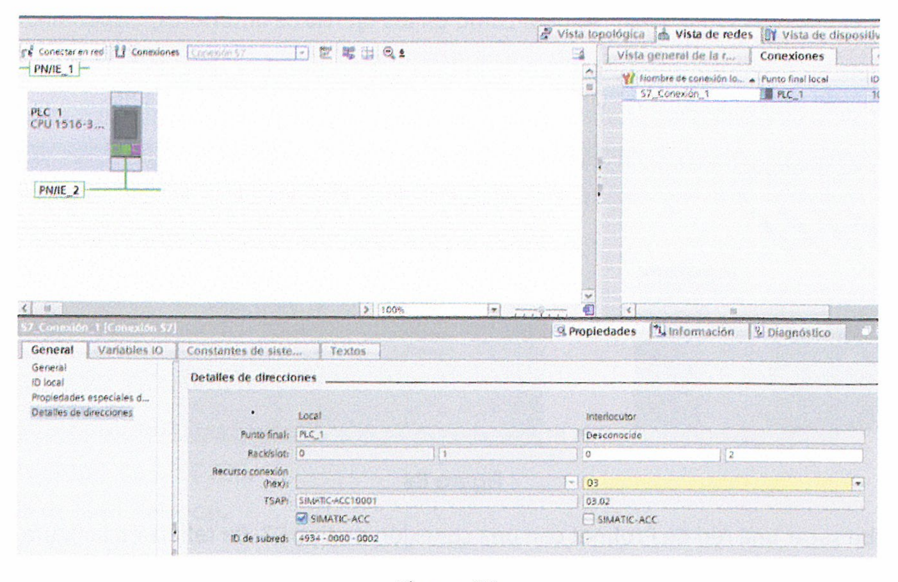


Figura 89

Ahora solo queda completar el OB1 del PLC 1500 y PLC 300.

En el PLC 1500, se colocarán los bloques GET y PUT y las órdenes para activar los respectivos PLC.

En la figura 90 puede verse el OB1 del PLC 1516 escrito en TIA PORTAL.

En la figura 91 se muestra el programa del OB1 realizado desde STEP 5.5.

```
CALL  PUT , "PUT_DB"                                %DB2
    Remote - Variant
    REQ    :="Clock_10Hz"                           %M0.0
    ID     :=W#16#100                               W#16#100
    DONE   :="Tag_1"                                %M1.0
    ERROR  :="Tag_2"                                %M1.1
    STATUS :="Tag_3"                                %MW2
    ADDR_1 :=P#M205.0 BYTE 1                        P#M205.0 BYT…
    ADDR_2 :=
    ADDR_3 :=
    ADDR_4 :=
    SD_1   :=P#M200.0 BYTE 1                        P#M200.0 BYT…
    SD_2   :=
    SD_3   :=
    SD_4   :=

CALL  GET , "GET_DB"                                %DB1
    Remote - Variant
    REQ    :="Clock_10Hz"                           %M0.0
    ID     :=W#16#100                               W#16#100
    NDR    :="Tag_7"                                %M4.0
    ERROR  :="Tag_8"                                %M4.1
    STATUS :="Tag_9"                                %MW5
    ADDR_1 :=P#M100.0 BYTE 1                        P#M100.0 BYT…
    ADDR_2 :=
    ADDR_3 :=
    ADDR_4 :=
    RD_1   :=P#M105.0 BYTE 1                        P#M105.0 BYT…
    RD_2   :=
    RD_3   :=
    RD_4   :=

U      "Tag_10"                                     %M105.0
=      "Tag_11"                                     %A0.0

U      "Tag_12"                                     %E0.0
=      "Tag_13"                                     %M200.0
```

Figura 90

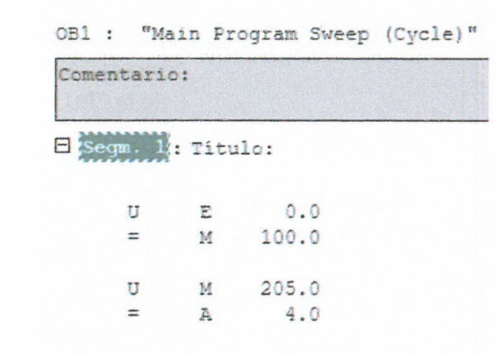


Figura 91

Ampliación sobre el TSAP

Cuando se realiza una conexión de forma automática, se asignan una serie de recursos adecuados a la configuración de la conexión. El TSAP (Transport Service Access Point) es la dirección de ese recurso. Cada recurso requiere formar su TSAP de manera correcta. Cuando se realizan conexiones que se encuentran en el mismo proyecto, el propio TIA PORTAL crea la dirección del recurso correcta (su propio TSAP). Por eso, en los ejercicios anteriores no se ha hecho uso del TSAP. En realidad, le ha dado su TSAP adecuado. En este caso, hay una estación que no se encuentra especificada en el proyecto y, por ello, se ofrece un TSAP que puede no ser correcto y deberá revisarse.

Para formar el TSAP, existen unas reglas específicas en función del tipo de conexión.

La estructura del TSAP es distinta dependiendo del equipo:

- Para CPU S7-1500:

  SIMATIC-ACC <nnn><mm>, siendo ***nnn*** la ID local y ***mm*** un valor libre.

- Para CPU S7-300/400:

  <xx>.<yz>, siendo ***xx*** el número del recurso de conexión, ***y*** el número de *rack* y ***z*** el número de *slot*.

La estructura del TSAP también será diferente dependiendo del interlocutor:

La estructura del TSAP en las CPU S7-1500 depende del interlocutor remoto. Si se conecta una CPU S7-1500 con una CPU S7-300/400, la primera utiliza también la estructura del TSAP e incluye el recurso de conexión.

Estos son algunos ejemplos para el TSAP de distintas configuraciones de conexiones:

- Conexión entre dos CPU S7-1500:
  - CPU S7-1500 A con ID local 200: TSAP: SIMATIC-ACC20001.
  - CPU S7-1500 B con ID local 3B2: TSAP: SIMATIC-ACC3B201.
- Conexión entre una CPU S7-1500 y una CPU S7-300/400:
  - CPU S7-1500 (*rack* 0, *slot* 1, recurso de conexión 10): TSAP: 10.01.
  - CPU S7-300/400 (*rack* 0, *slot* 2, recurso de conexión 11): TSAP: 11.02.

En cuanto a los rangos de valores del recurso de conexión,

La siguiente tabla informa acerca del significado de los valores y el tipo de conexión. Dependiendo del interlocutor y el tipo de conexión, el rango de valores se limita automáticamente a valores válidos, o el valor del recurso de conexión se asigna de forma fija.

Recurso de conexión (1.^er^ *byte* del TSAP)	Significado	Tipo	Significado
0x01 (PG)	Conexión PG	Conexión libre (no está configurada)	Se ha reservado, como mínimo, un recurso por CPU para conexiones PG. En el caso de determinadas CPU S7-300, se pueden reservar varios recursos en las propiedades de la CPU.
0x02 (OP)	Conexión OP	Conexión libre (no está configurada)	Se ha reservado, como mínimo, un recurso por CPU para conexiones OP. En el caso de determinadas CPU S7-300, se pueden reservar varios recursos en las propiedades de la CPU.
0x03	Otras	Conexión libre (conexión configurada no especificada)	Este recurso de conexión puede ofrecer servicio a varias conexiones. Utilización: ¡conexión unilateral para interlocutor no especificado! No se necesita configurar el interlocutor cuando se direcciona el recurso de conexión 0x03.
0x10... 0xDF	Conexiones con establecimiento de conexión estático o dinámico	Conexión configurada no especificada	Uno de estos recursos de conexión puede dar servicio exactamente a una conexión. Utilización: ¡conexión bilateral para interlocutor no especificado!

Conexiones S7 libres

Las conexiones S7 libres se establecen de forma dinámica durante el tiempo de funcionamiento; existen las siguientes conexiones libres:

- Conexiones PG (0x01)

  Para conexiones S7 establecidas normalmente por una programadora o un PC (con funcionalidad ES), la estación/el módulo direccionado se configura, se programa, se comprueba y se pone en funcionamiento mediante una conexión de este tipo; por regla general, la conexión vuelve a eliminarse a continuación. A través de este recurso de conexión resulta posible tanto la lectura como la escritura (p. ej., observar y cargar).

- Conexiones OP (0x02)

  Para conexiones S7 establecidas normalmente desde un OP o un PC (con funcionalidad OS), mediante una conexión de este tipo, se observa la estación/el módulo en referencia al proceso controlado.

- Otras (0x03)

  En los casos donde la utilización no haya sido fijada, durante la configuración de una conexión unilateral S7 desde un S7-400 hasta un S7-300, se utiliza este recurso de forma automática.

- Comunicación básica S7 (0xFD)

  Para las conexiones establecidas normalmente desde una CPU hasta otro módulo (CPU o FM) dentro de una subred, el establecimiento de conexión se inicia por el programa de aplicación y no existe ninguna configuración de conexión. A partir de una conexión de este tipo, se pueden intercambiar datos de proceso entre los módulos. En el caso de determinadas CPU S7-300, es posible reservar recursos para la comunicación básica S7.

Para la selección del recurso de conexión del interlocutor local, STEP 7 propone siempre un recurso de conexión libre. Se recomienda utilizar el recurso propuesto.

En este ejercicio, al tratarse de una conexión de una estación configurada pero no especificada, al interlocutor le corresponde un 03 en el primer *byte* del TSAP y, como se trata de un S7-300 colocado en el *slot* 2, el segundo *byte* será un 02. Por eso, el TSAP es 03.02, como puede verse en la figura 89.

7. Ejercicio de Profinet de un PLC S7-1500 y una CPU S7-300 no especificada, con CP

Enunciado

Este ejercicio constituye un caso que se puede presentar en tiempos de cambios y modificaciones en una industria. Resulta normal que las modificaciones y actualizaciones en una pequeña o mediana empresa se realicen de una forma escalonada. Por eso, es muy importante disponer de sistemas escalables y abiertos. Con el siguiente ejercicio, se trata de acomodar un sistema que dispone de un PLC anticuado a otro actualizado y moderno. Se sabe que determinadas CPU (antiguas y descatalogadas antes de octubre de 2007) no son compatibles con TIA PORTAL. Este caso se parece al ejercicio anterior, pero aquí el PLC no presenta la interfaz de Profinet integrada.

Se trata de comunicar por Profinet un PLC S7-1500 y un S7-300; concretamente, el PLC S7 1516 3PN/DP y el S7-315 2DP. Este PLC de la serie S7-300 constituye un autómata de esos no compatibles con TIA PORTAL, además de no disponer de interfaz de Profinet integrada. Por ello, se deberá colocar una CP 343 para posibilitar la conexión del autómata S7-315 con el S7-1516 por Profinet.

Igual que se hacía en el anterior ejercicio, el S7-1500 se va a configurar y programar en TIA PORTAL y el S7-300 utilizará el *software* clásico de STEP 7 V5.5.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una CPU 315 2DP con los siguientes módulos centralizados:
   - Una fuente de alimentación SITOP de 5 A.
   - Un módulo de 16 entradas digitales a 24 V (DC).
   - Un módulo de 16 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas y 2 salidas analógicas.
3. Una CP343-1 Advanced.
4. Un cable de Profinet para las conexiones.

Realización

El proceso de realización se muestra similar al anterior. La única novedad estriba en que se debe configurar la CP 343-1 Advanced. Todo esto se hará desde el STEP 7 V5.5. Lo referente al S7-1516 resulta idéntico a lo realizado en el ejercicio anterior.

Al final del ejercicio, se puede ampliar información respecto a esta CP 343-1 Advanced.

Esta CP se puede utilizar como controlador o como Device. Si se emplea como controlador, se comportará como la CPU; como de Device, lo hace como una estación descentralizada, es decir, como si fuera una ET.

Para este ejercicio se usará como controlador y se deberá configurar en uno de los *slots* del bastidor donde se halla el PLC.

La CP 343-1 se encuentra en el catálogo. Si no aparece en él, se debe cargar el fichero *GSD* de la CP 343-1. En la figura 92 se indica dónde se encuentra dentro del catálogo.

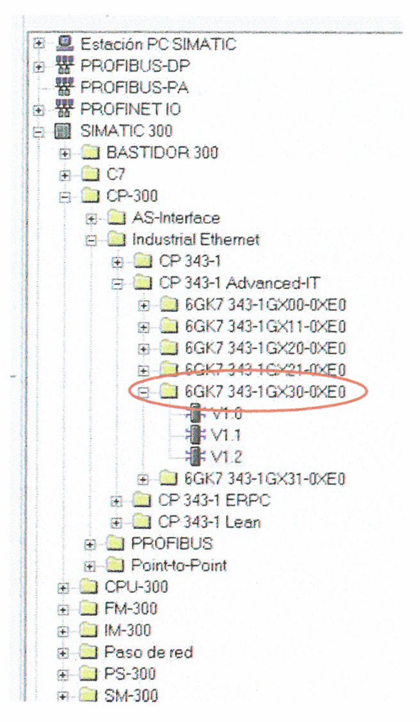


Figura 92

Una vez configurado el PLC junto a la CP, quedará del modo señalado en la figura 93.

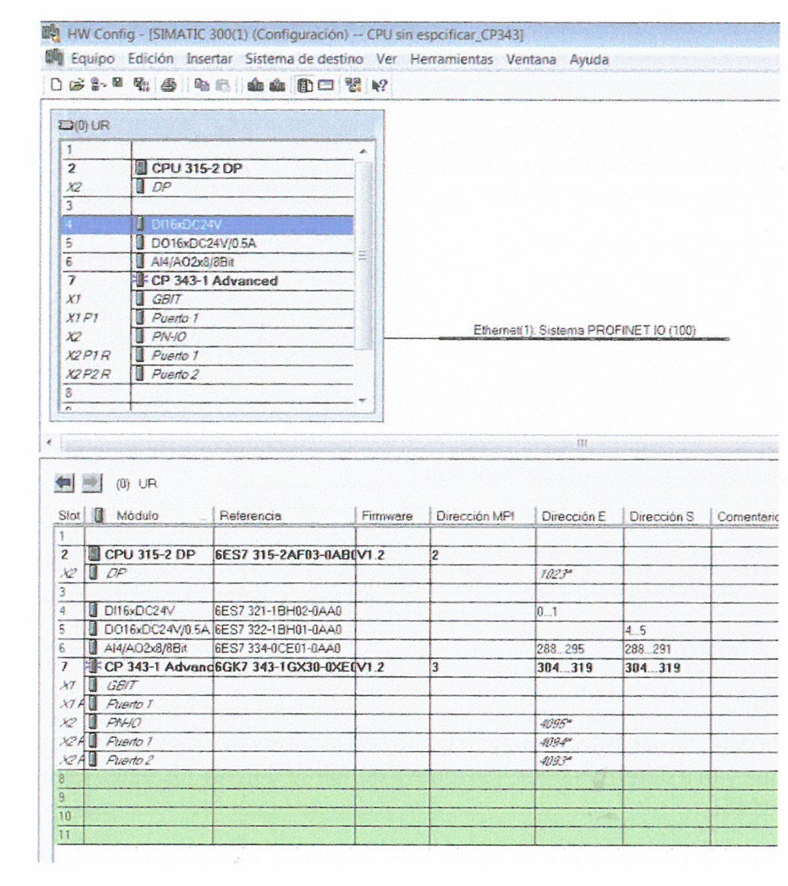


Figura 93

Para enviar esta configuración al PLC, la primera vez se debe mandar vía PC-Adapter (vía MPI). Tras esta primera configuración, ya se podrá hacer por Profinet y se prescindirá del PC-Adapter.

Se recuerda que, para que la configuración sea correcta, el nombre de la CP 343-1 en el proyecto debe ser el mismo que tiene cargado. Si no fuese así, habrá que editarlo y ponerlo igual en el proyecto (*offline*) que en real (*online*). Para ello, se puede utilizar la herramienta de edición del propio STEP 7 V5.5, ubicado en la ventana de *hardware*. En la figura 94 puede verse esta acción. Una vez encontrada la estación, se modifica el nombre y se envía a la CP 343-1 pulsando en *Asignar nombre*. En las figuras 95 y 96 se ilustra esta descripción.

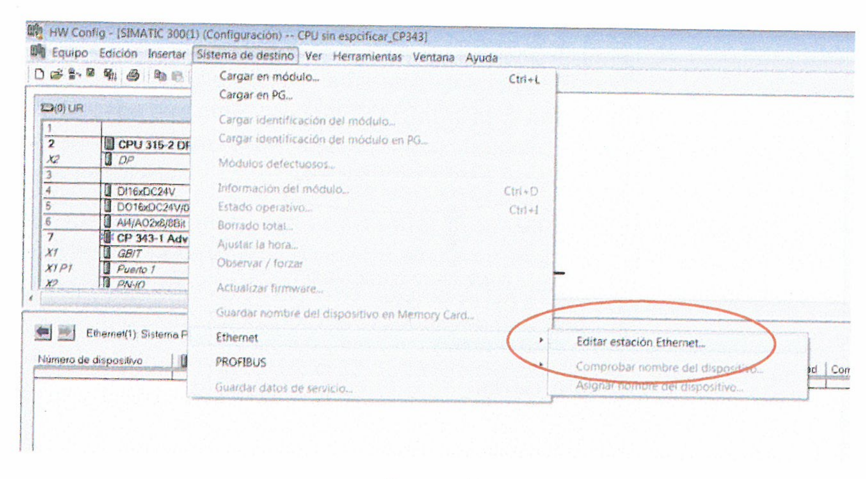


Figura 94

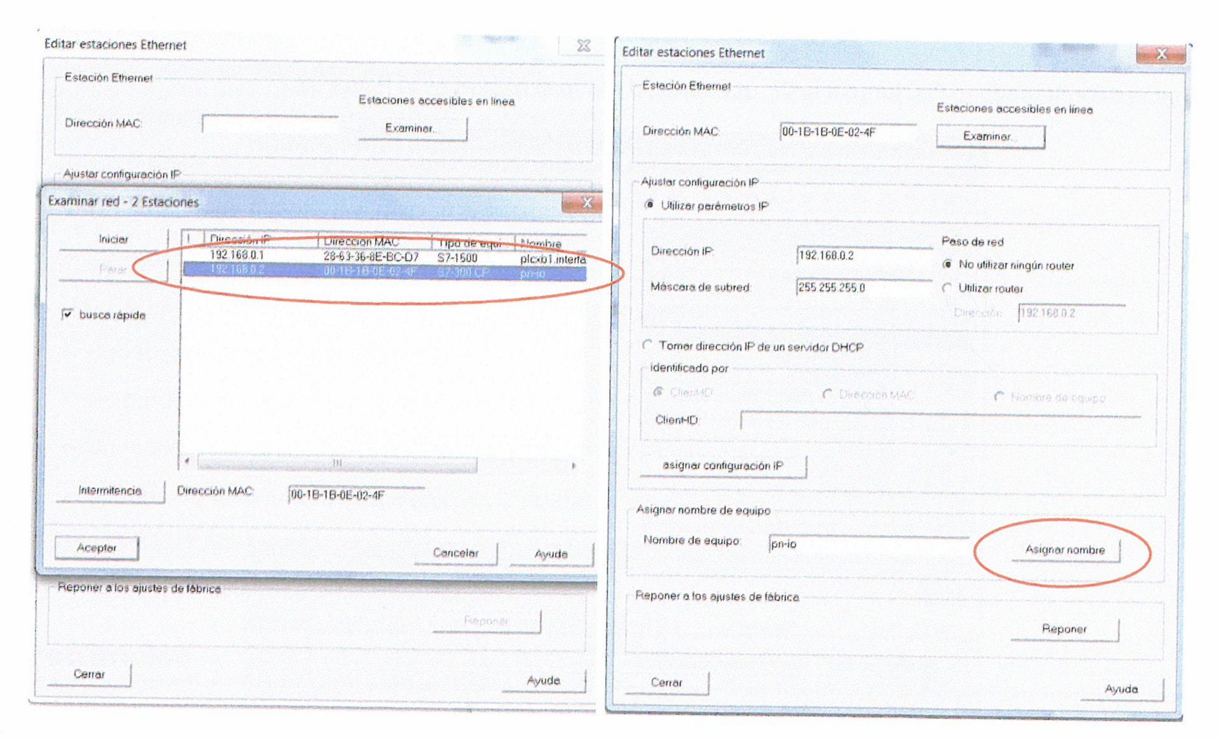


Figura 95

Figura 96

Comunicación de datos

Para completar el ejercicio, se necesita preparar a la CP para la comunicación de datos y, para ello, se han de usar dos funciones que van a determinar los «buzones» donde se dejarán los datos para intercambiar entre la CPU 315 y la CPU 1516.

Para intercambiar datos, se deben crear unos *buffers*, por donde se intercambiarán los datos con el PLC.

Como se ha dicho, se emplean dos funciones: una para recibir y otra para enviar. Estas funciones son la FC11 (PN I/O SEND), para enviar datos desde la CPU, y FC12 (PN I/O RECV), para recibir datos desde la CP 343-1.

El módulo CP se compone de una distribución de memoria que será una imagen del campo de direcciones de todas las E/S definidas en Profinet.

En el autómata habrá que seleccionar por dónde recibe y envía los datos hacia o desde la CP. Esa selección se realizará mediante marcas o DB con las funciones FC11 y FC12.

Se encuentran en la librería SIMATIC_NET_CP.

FC11 (PN I/O SEND)

El bloque FC PNIO_SEND se utiliza para la transmisión de datos en los modos de funcionamiento del CP Profinet IO-Controller o Profinet IO-Device, es decir, cuando hace función de PLC (Controller) o como estación de trabajo (Device).

Funcionamiento como Profinet IO-Controller

Este es el modo en que se utiliza en este ejercicio. El bloque transfiere los datos de procesos (salidas) de un área de salida indicada al CP, para su transmisión a los Profinet IO-Devices y proporciona como indicación de estado el *IO Consumer Status* (IOCS) de las salidas de los Profinet IO-Devices.

Funcionamiento como Profinet IO-Device

El bloque lee las entradas de procesos pretratadas de la CPU en el Profinet IO-Device y las transfiere al Profinet IO-Controller (direcciones configuradas); adicionalmente, el bloque proporciona como indicación de estado el *IO Consumer Status* (IOCS) del Profinet IO-Controller.

Los datos de proceso pretratados se ponen a disposición en un DB o en el sector de marcas. Se pueden utilizar CP en funcionamiento en paralelo de Profinet IO-Controller e IO-Device. Con el parámetro MODE, se ajusta el modo de funcionamiento para el que se debe llamar el FC.

Se debe conocer la dirección del módulo CP 343-1. Esto se puede ver en la ventana de «Hardware» y corresponde a la **primera dirección dada al módulo**. En la figura 97 se representa esta situación.

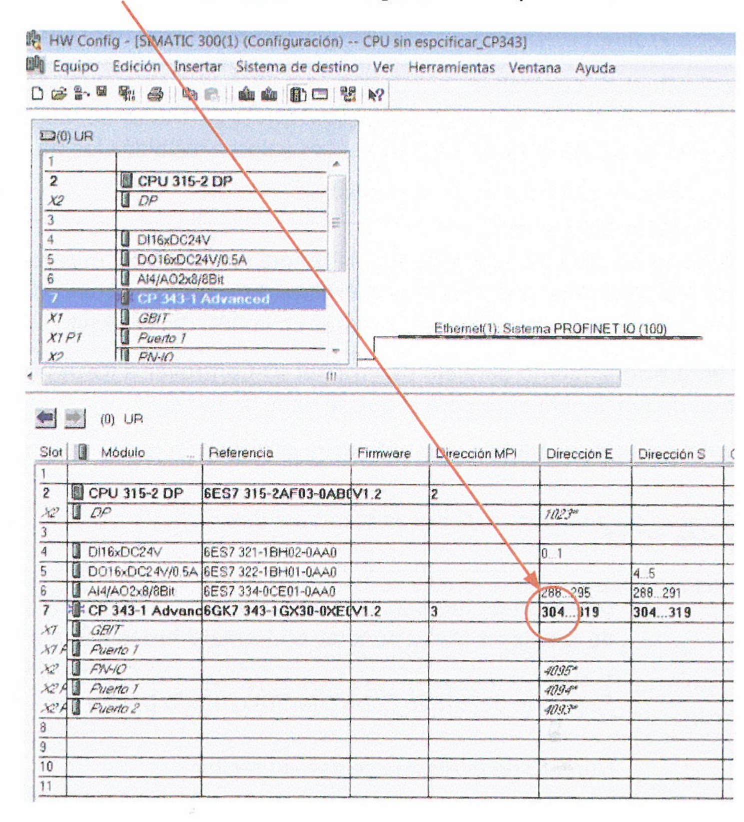


Figura 97

Parámetros de la función FC 11

CPLADDR:=	Dirección de módulo, como se indica en la figura 97, de la configuración de *hardware* (debe ponerse en formato hexadecimal)
MODE :=	Selección del modo Controller (0) o modo Device (1)
LEN :=	Número de *bytes* que se desean enviar
IOCS :=	IO Consumer Status: por cada *byte* de datos de emisión, se transmite un bit de estado. Como mínimo, hemos de colocar dos *bytes*. Se indica en formato puntero. Se pueden usar DB o marcas
DONE :=	Dirección para parámetro de retorno DONE
ERROR :=	Dirección para parámetro de retorno ERROR
STATUS :=	Dirección para parámetro de retorno STATUS
CHECK_IOCS :=	Dirección para parámetro de retorno CHECK_IOCS
SEND :=	Área de datos que se transmiten. Deben coincidir en número de *bytes* aquí y en el parámetro LEN

FC12 (PN I/O RECV)
El bloque FC PNIO_RECV se utiliza para la adopción de datos en los modos de funcionamiento del CP Profinet IO-Controller o Profinet IO-Device.

Funcionamiento como Profinet IO-Controller
El bloque transfiere los datos de proceso de los Profinet IO-Devices (entradas del Controller). También envía el *IO Provider Status (IOPS)* de los Profinet IO-Devices. Los valores recibidos los transfiere a las áreas de entrada indicadas.

Funcionamiento como Profinet IO-Device
El bloque adopta los datos de proceso transmitidos por el Profinet IO-Controller (direcciones de salida configuradas). También recibe *el IO Provider Status (IOPS)* del Profinet IO-Controller. Todos estos datos los escribe en las áreas de datos reservadas para las salidas de procesos en la CPU del Profinet IO-Device.

Se pueden utilizar CP en funcionamiento paralelo de Profinet IO-Controller e IO-Device. Con el parámetro MODE, se ajusta el modo de funcionamiento para el que se debe llamar el FC.

Resulta evidente que la dirección del módulo que se solicita en esta función será la misma que la de la función FC 11.

Parámetros de la función FC 12

CPLADDR:=	Dirección de módulo, como se indica en la figura 97, de la configuración de *hardware* (debe ponerse en formato hexadecimal)
MODE :=	Selección del modo Controller (0) o modo Device (1)
LEN :=	Número de *bytes* que se desean recibir
IOPS :=	IO Provider Status: por cada *byte* de datos de recepción, se transmite un bit de estado. Como mínimo hemos de colocar 2 *bytes*. Se indica en formato puntero. Se pueden usar DB o marcas
NDR :=	Dirección para parámetro de retorno NDR
ERROR :=	Dirección para parámetro de retorno ERROR
STATUS :=	Dirección para parámetro de retorno STATUS
CHECK_IOPS :=	Dirección para parámetro de retorno CHECK_IOPS
ADD_INFO :=	Información de diagnóstico
RECV :=	Área de datos para recibir. Deben coincidir en número de *bytes* aquí y en el parámetro LEN

* Para ampliar información sobre los comandos de FC11 y FC12, se puede revisar el siguiente documento (se necesita disponer previamente de una cuenta en la web de SIEMENS):

https://support.industry.siemens.com/cs/document/62543517/simatic-net-bloques-de-programa-para-cps-s7-simatic-net-manual-de-programaci%C3%B3n?dti=0&lc=es-WW

Función FC11 y FC12 para este ejercicio
A continuación, se van a implementar los parámetros de las dos funciones para este ejercicio.

FC11 (enviar datos)

CPLADDR:=W#16#130	Dirección de módulo, como se indica en la figura 97, de la configuración de *hardware* (debe ponerse en formato hexadecimal). La dirección que indica el *hardware* es 304 que, en hexadecimal, es 130.
MODE :=B#16#0	Selección del modo Controller (0). En hexadecimal.
LEN :=4	Número de *bytes* que se desean enviar; cuatro, en este caso.
IOCS :=P#M 125.0 BYTE 2	En este caso se indica con marcas. Se puede poner cualquiera, pero no podrán utilizarse para nada más. El formato debe ser puntero.
DONE :=M128.0	Cualquier marca es válida. El tipo de dato debe ser bit.
ERROR :=M128.1	Cualquier marca es válida. El tipo de dato debe ser bit.
STATUS := MW129	Cualquier marca es válida. El tipo de dato debe ser palabra (W).
CHECK_IOCS := M70.2	Cualquier marca es válida. El tipo de dato debe ser bit.
SEND := P#M 205.0 BYTE 4	Área de datos que se transmiten. Deben coincidir el número de *bytes* aquí y en el parámetro LEN. El formato ha de ser puntero.

FC12 (recibir datos)

CPLADDR:=W#16#130	Dirección de módulo, como se indica en la figura 97, de la configuración de *hardware* (debe ponerse en formato hexadecimal). La dirección que indica el *hardware* es 304 que, en hexadecimal, es 130.

MODE := B#16#0	Selección del modo Controller (0). En hexadecimal.
LEN :=4	Número de *bytes* que se desean recibir; cuatro, en este caso.
IOPS := P#M 60.0 BYTE 2	En este caso se indica con marcas. Se puede poner cualquiera, pero no podrán utilizarse para nada más. El formato debe ser puntero.
NDR := M74.7	Cualquier marca es válida. El tipo de dato debe ser bit.
ERROR := M74.5	Cualquier marca es válida. El tipo de dato debe ser bit.
STATUS := MW78	Cualquier marca es válida. El tipo de dato debe ser palabra (W).
CHECK_IOPS := M74.4	Cualquier marca es válida. El tipo de dato debe ser bit.
ADD_INFO := MW81	Cualquier marca es válida. El tipo de dato debe ser palabra (W).
RECV := P#M 100.0 BYTE 4	Área de datos que se transmiten. Deben coincidir en número de *bytes* aquí y en el parámetro LEN. El formato debe ser puntero. En este caso, se indica con marcas.

Según estos datos introducidos en las funciones FC11 y FC12, se puede afirmar que la CPU 315 recibe los datos de la periferia de Profinet a través de las marcas MB205, MB206, MB207 y MB208 (4 *bytes*). Igualmente, la CPU envía a la periferia de Profinet a través de las marcas MB100, MB101, MB102 y MB103 (4 *bytes*).

Resulta muy importante, en este caso, que las marcas elegidas para las funciones FC11 y FC12 coincidan con las que envía el PLC 1516. Como la configuración del PLC 1516 no se ha modificado con respecto al ejercicio anterior, se observa que el interlocutor envía por la MB100 y recibe por la MB205. En este caso, el interlocutor es el PLC 315+DP343 configurado y no especificado. En la figura 98 se indica esta situación.

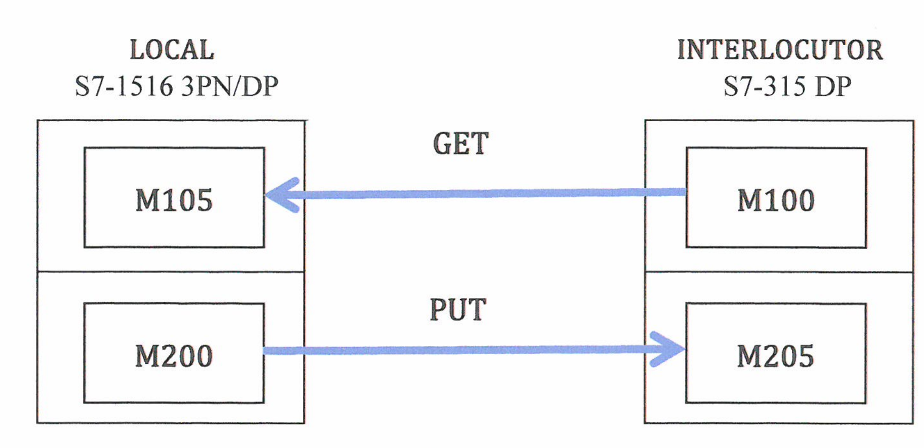


Figura 98

Para terminar el ejercicio, se colocan los dos OB1 del PLC S7-315 2DP y del S7-1516 3PN/DP.

OB1 del S7-315 2DP

```
⊟ Segm. 1: Titulo:
      CALL  "PNIO_SEND"                    FC11              -- PNIO SEND
       CPLADDR   :=W#16#130
       MODE      :=B#16#0
       LEN       :=4
       IOCS      :=P#M 125.0 BYTE 2
       DONE      :=M128.0
       ERROR     :=M128.1
       STATUS    :=MW129
       CHECK_IOCS:=M70.2
       SEND      :=P#M 205.0 BYTE 4

      CALL  "PNIO_RECV"                    FC12              -- PNIO RECEIVE
       CPLADDR   :=W#16#130
       MODE      :=B#16#0
       LEN       :=4
       IOPS      :=M74.6
       NDR       :=M74.7
       ERROR     :=M74.5
       STATUS    :=MW78
       CHECK_IOPS:=M74.4
       ADD_INFO  :=MW81
       RECV      :=P#M 100.0 BYTE 4

      U     M    205.0
      =     A      4.0

      U     E      0.0
      =     M    100.0
```

OB1 del S7- 1516 3PN/DP

Segmento 1: .....

Comentario

```
      CALL  PUT , "PUT_DB"                                   %DB2
         Remote - Variant
         REQ     :="Clock_10Hz"                              %M0.0
         ID      :=W#16#100                                  W#16#100
         DONE    :="Tag_1"                                   %M1.0
         ERROR   :="Tag_2"                                   %M1.1
         STATUS  :="Tag_3"                                   %MW2
         ADDR_1  :=P#M205.0 BYTE 1                           P#M205.0 BYT...
         ADDR_2  :=
         ADDR_3  :=
         ADDR_4  :=
         SD_1    :=P#M200.0 BYTE 1                           P#M200.0 BYT...
         SD_2    :=
         SD_3    :=
         SD_4    :=

      CALL  GET , "GET_DB"                                   %DB1
         Remote - Variant
         REQ     :="Clock_10Hz"                              %M0.0
         ID      :=W#16#100                                  W#16#100
         NDR     :="Tag_7"                                   %M4.0
         ERROR   :="Tag_8"                                   %M4.1
         STATUS  :="Tag_9"                                   %MW5
         ADDR_1  :=P#M100.0 BYTE 1                           P#M100.0 BYT...
         ADDR_2  :=
         ADDR_3  :=
         ADDR_4  :=
         RD_1    :=P#M105.0 BYTE 1                           P#M105.0 BYT...
         RD_2    :=
         RD_3    :=
         RD_4    :=

      U      "Tag_10"                                        %M105.0
      =      "Tag_11"                                        %A0.0

      U      "Tag_12"                                        %E0.0
      =      "Tag_13"                                        %M200.0
```

Ejercicio propuesto

En el ejercicio que se acaba de realizar, no existe periferia descentralizada. Se propone añadir una ET 200 al PLC 315 2DP y diseñar un programa en el que todos los dispositivos puedan enviar y recibir datos de los demás.

CP343-1 Advanced
Este módulo se encarga de conectar las CPU de la serie S7-300 que no dispongan de conexión Profinet a dicha red.

Además, cuenta con otras cualidades relacionadas con las IT (tecnologías de la información) (figura 99).

El CP soporta:

- Comunicación PG/OP.
- Comunicación S7.
- Comunicación abierta (SEND/RECEIVE).
- Comunicación Profinet.
- Comunicación IT.
- Funciones de seguridad, *firewall* y VPN.

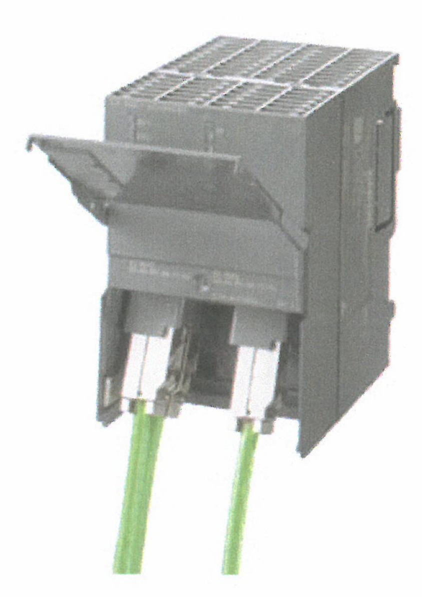


Figura 99

El siguiente texto se extrae de la WEB de SIEMENS y en él se establecen las características de la CP 343 Advanced:

http://w3.siemens.com/mcms/industrial-communication/es/ie/conexiones-de-sistema/simatic-s7-sinumerik-o/s7-300/Pages/cp343-1advanced.aspx

En los módulos de comunicaciones S7-Advanced se aprovechan las ventajas de Industrial Ethernet respecto al bus de campo. A través de esos módulos se puede establecer la comunicación entre el nivel de campo y el entorno TI sin necesidad de pasarelas (gateways) *intercaladas. Los Advanced CP posibilitan así una estructura de comunicaciones plana con poco esfuerzo de configuración. En muchos casos se puede prescindir de pasarelas* (gateways) *de comunicación entre nivel de campo, MES y nivel ERP, con lo que se reduce el esfuerzo de configuración y el número de fuentes de error.*

Servicios estándar para la producción
El CP 343-1 Advanced ofrece una comunicación eficiente con S5, S7 y con servidores OPC o con la PG, al igual que con el CP 343-1. La comunicación puede configurarse también a través de TCP, lo que permite la conexión de diversos sistemas.

Profinet
Como controlador Profinet, el CP asume igualmente en la máquina el control de los módulos de entrada y salida descentralizados. El CP soporta a través de Profinet CBA esquemas de instalación modulares orientados a objetos.

Comunicación TI
Posibilidad de transmitir por correo electrónico información importante sobre datos de producción. Por ejemplo, el departamento de aseguramiento de la calidad puede someter a análisis datos de máquinas actualizados, todo ello en sitios web. Con el protocolo FT, el controlador puede importar a través del CP directamente datos de un mainframe *o recoger archivos de PC para ejecutar una petición.*

El tema de la seguridad
El CP 343-1 Advanced ofrece gran robustez frente a ataques desde la red a pesar de que domina las comunicaciones TI. Eso hace innecesarias las actualizaciones de seguridad constantes. Una lista de IP protege del acceso por parte de PC no autorizados. Un sistema de seguridad básica, a través de contraseñas y asociado a personas y servicios de comunicación, protege de accesos no autorizados.

Soporte ideal para el mantenimiento
El servicio de mantenimiento permanece informado por correo electrónico sobre fallos y perturbaciones desde cualquier lugar. El diagnóstico remoto se lleva a cabo en sitios web integrados que señalizan el estado del controlador. Los fallos de la máquina pueden diagnosticarse online *con un navegador de Internet mediante sitios web de creación propia. Hasta la documentación de una máquina puede estructurarse de forma conveniente vía HTML y archivarse en el servidor web del CP. El módulo puede sustituirse sin necesidad de herramientas de configuración, pues el sistema de archivos del servidor web se encuentra en el C-Plug enchufable.*

Separación de redes entre nivel de campo y nivel de empresa
A menudo las redes se separan entre sí físicamente, pero deben comunicarse entre sí. Motivos para separar tales redes son lograr desacoplamientos selectivos de cargas o independizar diferentes áreas dentro de una empresa (p. ej., la red de oficina y de fabricación). Si se usa el procesador de comunicaciones CP 343-1 Advanced en controladores S7, este requisito no supone un problema. Con la introducción de conexiones para subredes IP separadas en Gigabit Ethernet y FastEthernet en un módulo, es posible el uso salvando límites de red de servicios TI mediante Routing IP estático. El acceso protegido al PLC y el tráfico de datos salvando límites de red se regula a través de una lista de acceso IP configurable.

8. Ejercicio de AS-i/DP con un PLC S7-1500

Enunciado

La red de AS-i resulta extremadamente cómoda por su sencillez. En este ejercicio se va a comunicar un PLC S7-1516 3PN/DP con una red de AS-i. Para ello, se empleará una pasarela (*gateway*) que haga de maestro de AS-i y esclavo de Profibus o controlador de Profinet y maestro de AS-i. En este ejercicio utilizará el DP/AS-i LINK 20E de Siemens; concretamente, el de referencia 6GK1415-2AA01. Es una pasarela maestro de AS-i y esclavo de Profibus. También valdría el de referencia 6GK1415-2AA10. Igualmente se podría utilizar otra pasarela que conectara AS-i con Profinet.

En la red de AS-i, se dispondrá de un grupo de esclavos que intercambiarán información con el PLC S7-1500 y toda su periferia.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Un LINK DP/AS-i LINK 20E de referencia 6GK1415-2AA01 y diversos esclavos E/S de AS-i.
3. Un cable de Profinet y Profibus para las conexiones.

Realización

En primer lugar, se deberá configurar el PLC S7-1516 como se ha hecho hasta ahora. Para realizar este ejercicio, se ha de disponer de un PLC con interfaz de Profibus integrado, ya que el LINK empleado es un esclavo de Profibus. Si no se dispusiera de esa conexión de Profibus, habría que utilizar otro LINK que conectara Profinet y AS-i. El PLC S7-1516 cuenta con esa interfaz integrado.

A continuación, se debe ir a *Vista de redes* y buscar en el catálogo el LINK correspondiente. Si el TIA PORTAL se halla actualizado, deberá encontrarse, pero, si no fuese así, se instalaría su fichero GSD. En esta ruta se puede bajar (recuerde que hay que estar dado de alta en la página de Siemens): https://support.industry.siemens.com/cs/es/es/view/113250.

Una vez colocado el Link 20E dentro de *Vista de redes*, quedará como aparece en la figura 100.

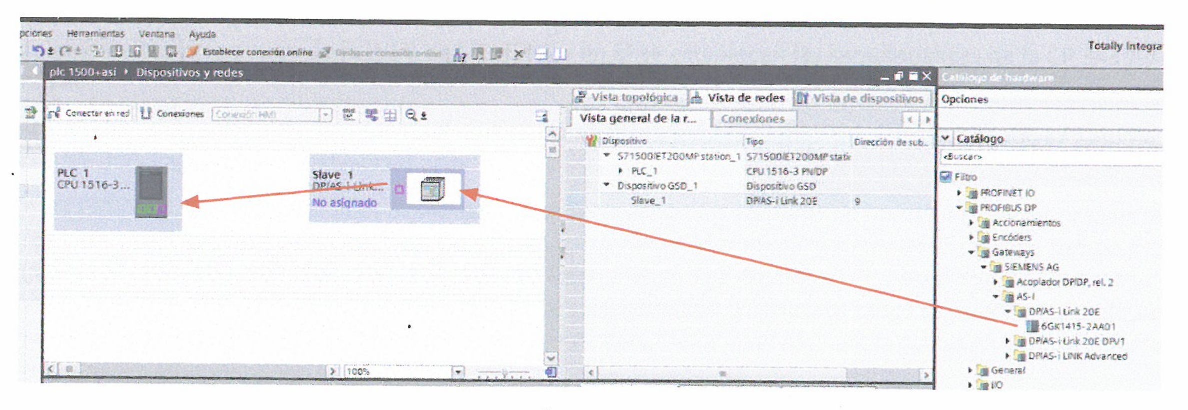


Figura 100

Seguidamente, se realiza la conexión de Profibus entre los dos, uniendo los dos cuadraditos morados. Permanecerán unidos el Link y el PLC y, a la vez, el Link queda asignado al PLC_1, como se indica en la figura 101.

Unidad 4 · Ejercicios prácticos de Profinet

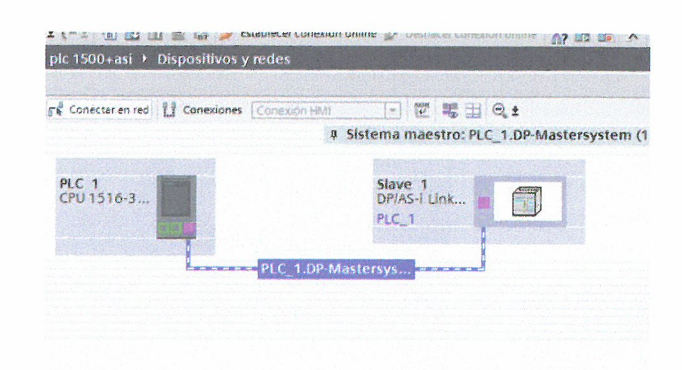


Figura 101

Ahora se debe abrir el Link 20E en la vista de dispositivos y añadir uno de los bloques que aparecen en el catálogo. Véase la figura 102, donde se indica este paso.

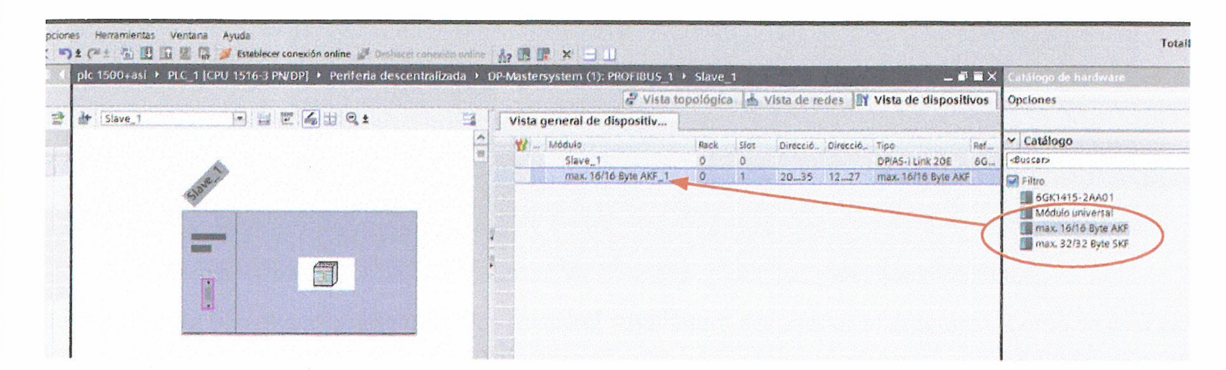


Figura 102

Si se coloca el *Módulo universal* en las propiedades del módulo (parte inferior), se debe seleccionar alguna de las opciones que salen, como «Entrada», como «Salida» o como «E/S». Esto se indica en la figura 103.

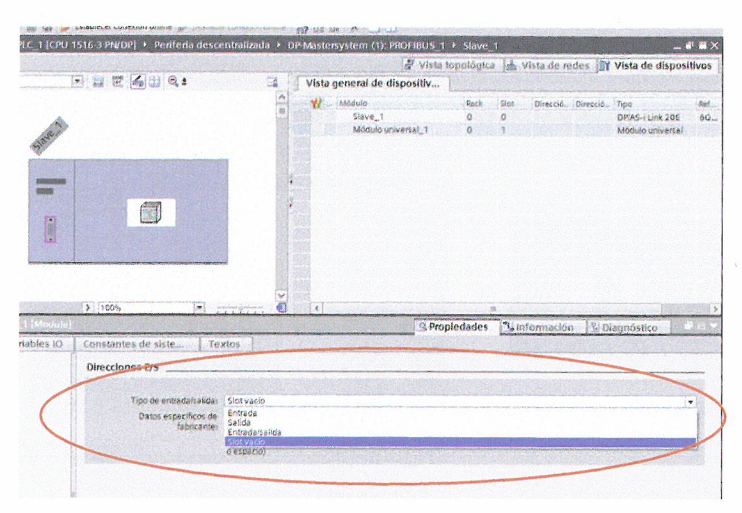


Figura 103

Si se coloca cualquiera de los otros dos módulos, por defecto sale como «E/S» y no se puede cambiar. Véase la figura 104, donde se ha seleccionado un módulo AKF.

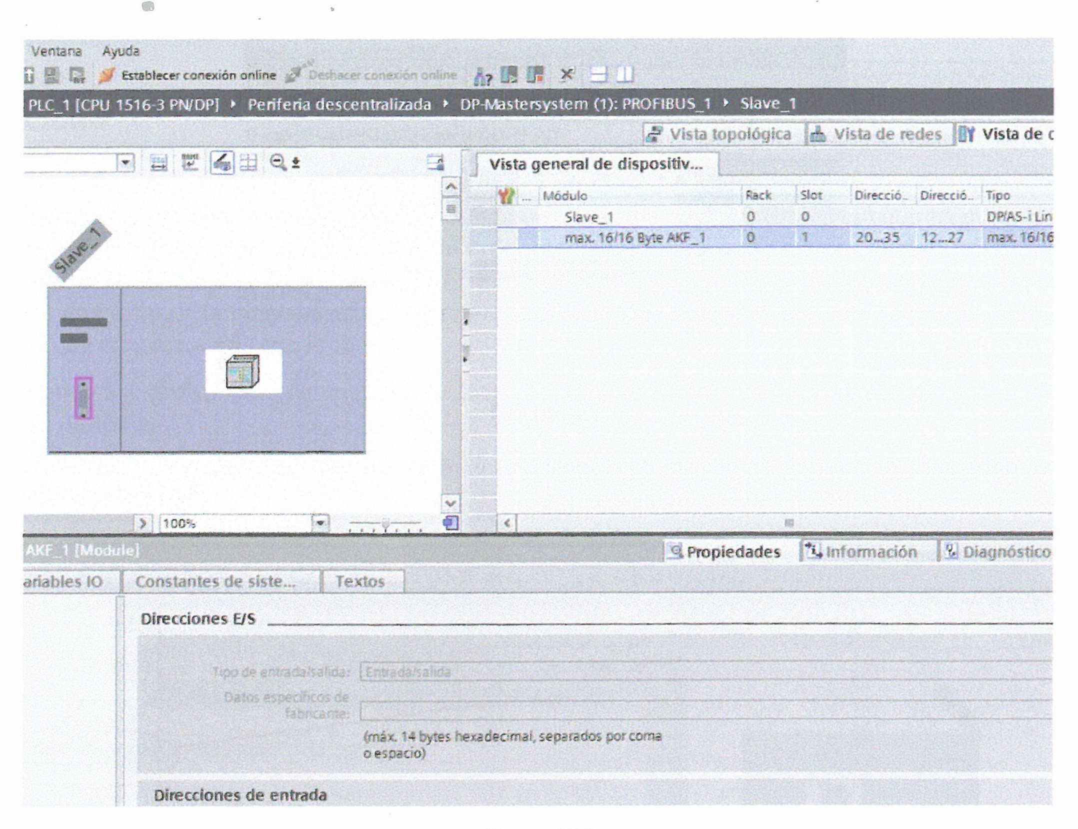


Figura 104

Por comodidad, se debe cambiar las direcciones de entrada y salida y ponerlas igual en las dos. Este paso no resulta necesario, pero es cómodo. En este caso, se pone la dirección 40. Debe tratarse de una dirección compatible con la entrada y la salida. Para cambiar, se pulsa dos veces sobre las direcciones y, pasa a 40 y se acepta. Se indica en la figura 105.

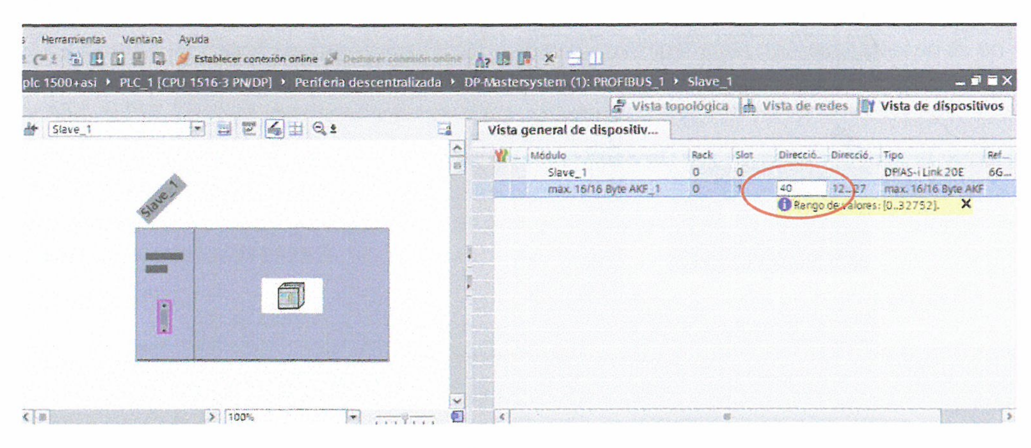


Figura 105

Si, en la comprobación del funcionamiento, no establece comunicación entre las E/S del PLC y los módulos de E/S AS-i, **habrá que cambiar los módulos (universal, AKF, SKF) del LINK 20E hasta que alguno de ellos funcione.**

Disponer de un módulo universal (número de entradas/salidas seleccionable por usuario), AKF *(General Identification Format)* o SKF *(Special Identification Format)* constituye la forma de reservar la memoria de direcciones de intercambio con Profibus.

Una vez configurado el Link 20E, se ha de hacer lo mismo con los distintos dispositivos de E/S AS-i que se dispongan. Para ello, existen dos formas de realizarlo: una consiste en incluir en el proyecto de TIA PORTAL dichos dispositivos; la otra es no hacerlo y, para direccionar cada dispositivo, se emplea el mapeo de direcciones que ejecuta AS-i. En este ejercicio utilizará el segundo método ya que resulta más práctico.

En primer lugar, se debe asignar a cada unidad de E/S AS-i una dirección de AS-i formada por un número entre 1 y 31, lo cual se realiza expresamente con la programadora disponible para ello. En este caso, se dispone de una columna de emergencia, una caja de pulsadores, un detector inductivo y un módulo de dos contactores, como se ve en la figura 106. En dicha figura se muestra también impresa la dirección de los esclavos de AS-i. La columna tiene una dirección 12; los contactores, dirección 13; la caja de pulsadores, dirección 14 ,y el detector, dirección 4.



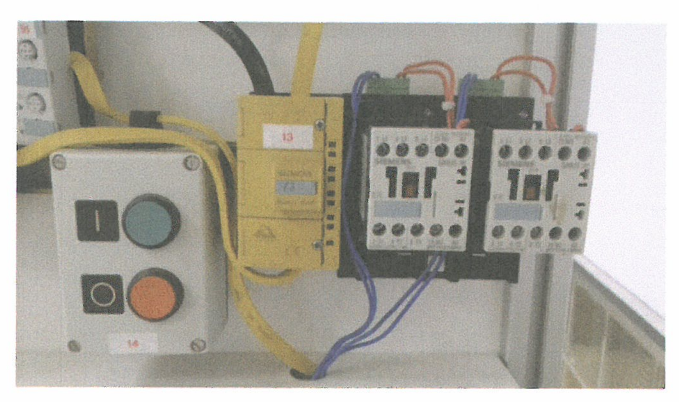


Figura 106

Una vez que se han direccionado todos los esclavos, solo faltará desarrollar el programa. Como se va a utilizar el método de no configurar los dispositivos dentro de TIA PORTAL, se acudirá a la tabla de direcciones que presenta el propio CP 343-1. Para ello, se debe conocer la primera dirección de entradas y de salidas que ha dispuesto la configuración cuando se ha colocado el CP 343-1 en el proyecto.

En la figura 107 se ve dónde se pueden encontrar las **direcciones**. Desde la vista de dispositivo, se distinguen. Se habían cambiado las direcciones para que ambas, tanto la entrada como la salida, fueran iguales.

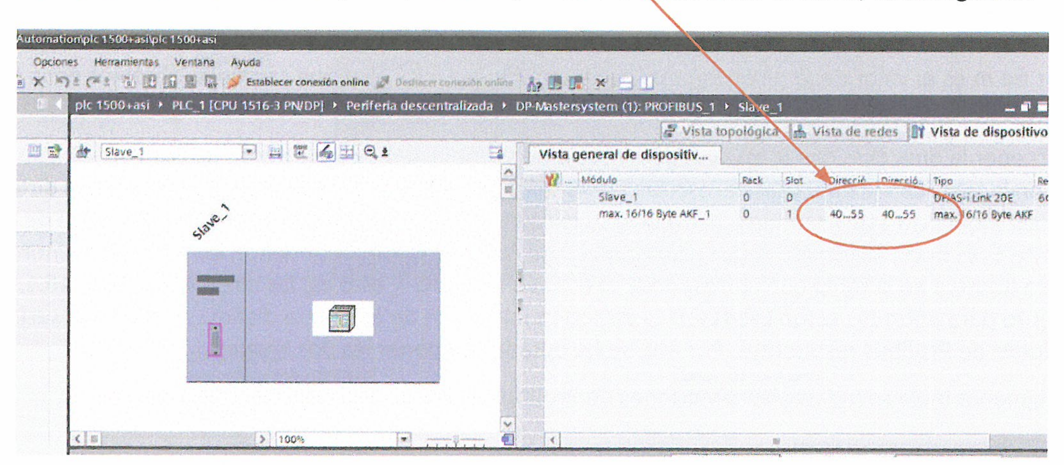


Figura 107

Se toma la primera dirección; en este caso, la 40. Con esa dirección se entra en la tabla comentada y se extrae la dirección, que habrá que colocar en el programa para acceder al dispositivo.

A cada esclavo de AS-i, el Link 20E le asigna 4 bits para la dirección de entrada o salida. Se reparte entre los cuatro bits altos y cuatro bits bajos. El Link 20E realiza un mapeado de memoria para los 32 posibles esclavos de AS-i. Con la dirección que se extrae de la tabla, el maestro de Profibus (el PLC S7-1516 en este caso) accederá a los distintos dispositivos de AS-i.

La tabla es la siguiente:

	Bit 4...7 (Bit 7, Bit 6, Bit 5, Bit 4)	Bit 3...0 (Bit 3, Bit 2, Bit 1, Bit 0)
m+0		Esclavo 1
m+1	Esclavo 2	Esclavo 3
m+2	Esclavo 4	Esclavo 5
m+3	Esclavo 6	Esclavo 7
m+4	Esclavo 8	Esclavo 9
m+5	Esclavo 10	Esclavo 11
m+6	Esclavo 12	Esclavo 13
m+7	Esclavo 14	Esclavo 15
m+8	Esclavo 16	Esclavo 17
m+9	Esclavo 18	Esclavo 19
m+10	Esclavo 20	Esclavo 21
m+11	Esclavo 22	Esclavo 23
m+12	Esclavo 24	Esclavo 25
m+13	Esclavo 26	Esclavo 27
m+14	Esclavo 28	Esclavo 29
m+15	Esclavo 30	Esclavo 31

El valor de *m* es el valor de la dirección que se le ha dado al Link 20E y que se puede observar en la figura 107 y que, en este ejemplo, era 40.

Para obtener la dirección con la que el PLC maestro de Profibus debe encontrar al dispositivo AS-i, se debe entrar en la tabla con la dirección del esclavo de AS-i. Si, por ejemplo, el esclavo de AS-i posee una dirección 13 (en este ejercicio esa dirección corresponde a los contactores AS-i), le pertenece una dirección de Profibus de *m+6;* es decir, 40+6= 46. Sería la dirección 40 de Profibus y, concretamente, los 4 bits de menor peso, como se observa en la tabla que corresponde al esclavo 12 de AS-i; esto es, las direcciones 46.0, 46.1, 46.2, 46.3 tanto para entradas como salidas. Si la dirección inicial (*m*) de E/S fuera distinta, también lo serían las direcciones de Profibus resultantes. Por eso, se ha optado por poner las dos iguales.

En la siguiente tabla se indican las direcciones de Profibus de los dispositivos de este ejemplo:

Contactores	Esclavo 13	m+6 40+6=46	*Bytes* bajos 46.3-46.2-46.1-46.0
Pulsadores	Esclavo 14	m+7 40+7=47	*Bytes* altos 47.7-47.6-47.5-47.4
Columna de señalización	Esclavo 12	m+6 40+6=46	*Bytes* altos 46.7-46.6-46.5-46.4
Detector inductivo	Esclavo 4	m+2 40+2=42	*Bytes* altos 42.7-42.6-42.5-42.4

Como solo existen dos contactores, solo se emplean 2 bits de los cuatro posibles. Los 2 bits primeros (46.0 y 46.1) serán los de los contactores. Para saber cuál es cuál, habrá que probar.

El OB1 podría ser así:

U E0.0 = A 46.0	Una entrada del PLC activa el esclavo de AS-i contactor.
U E 47.4 = A4.0	Un pulsador esclavo de AS-i activa una salida del PLC.
U E 42.4 = A46.4	El esclavo de AS-i sensor inductivo activa una lámpara del esclavo de AS-i de señalización.

DP/AS-i Link 20E

Las figuras que aparecen en este apartado pertenecen al manual del módulo DP/AS-i Link 20E, de Siemens, que puede bajarse de la siguiente dirección (hay que darse de alta en la web de Siemens):

https://cache.industry.siemens.com/dl/files/638/5281638/att_42352/v1/Link20E_s.pdf

Este módulo es una *gateway* que permite el intercambio de *bytes* y palabras de datos de E/S entre un maestro de Profibus y los esclavos de AS–i.

En la figura 108 se distingue el frontal del Link 20E, así como las funciones de cada elemento.

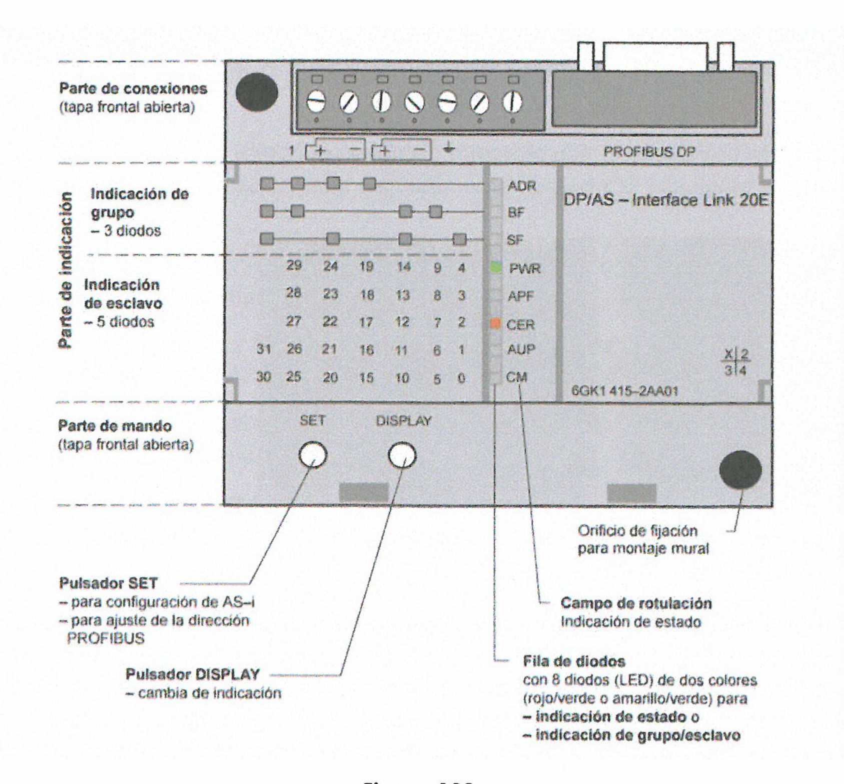


Figura 108

Con el módulo, se puede saber qué direcciones de esclavos de AS-i existen, de forma que se asegure que todos los esclavos sean reconocidos. No se puede saber de quién es cada dirección. Para conocerlo, se emplea la programadora. Para leer las direcciones, se utilizan los 8 diodos ledes del frontal. Los tres primeros ledes (ADR, BF y SF) indican en qué grupo se debe leer la dirección; es decir, en qué columna se debe fijar la atención para determinar su dirección. Los otros 5 ledes indicarán si existe (led verde encendido) o no (led apagado) dirección en esa columna examinada. Para pasar de grupo, esto es, de columna, se pulsará en el botón DISPLAY. En las figuras 109-116 se representa la forma de proceder desde el punto de partida hasta el

final. Constituye un ejemplo real donde se presentan una serie de esclavos con sus respectivas direcciones de AS-i que se van a detectar.

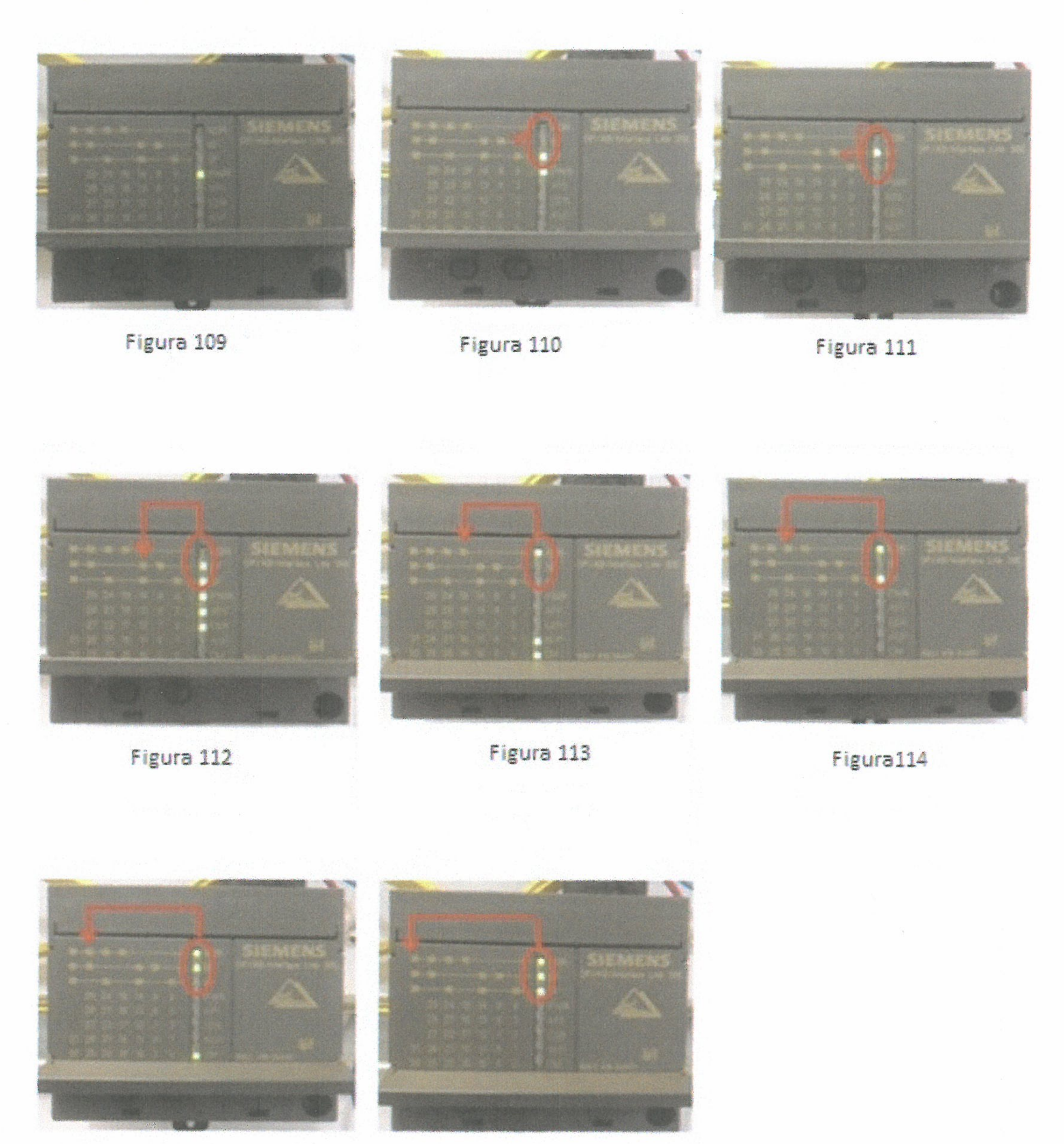


Figura 109 Figura 110 Figura 111

Figura 112 Figura 113 Figura114

Figura 115 Figura 116

La figura 109 es el punto de partida donde se indica que está alimentado *(Power)* y no hay errores. Pulsando una vez sobre DISPLAY, se pasa a la figura 110, donde los tres primeros bits muestran la columna donde se debe fijar la atención. Se aprecia que de los cinco ledes restantes, uno se encuentra encendido. Leyendo en la columna actual, se observa que la dirección que existe es la 4. En la figura 111 no se señala ningún led encendido de los cinco restantes, por lo que no existe ninguna dirección de las que indica dicha columna. Lo mismo ocurre con la figura 114 y la 116. En las figuras 112, 113 y 115 sí que se marcan direcciones. En la figura 112 se observan las direcciones 12, 13 y 14; en la figura 113, las direcciones 15 y 16 y, en la figura 115, la dirección 25.

Como este módulo es un esclavo de Profibus, se le debe aportar una dirección. La dirección de Profibus del módulo se lee cuando el led ADR se queda en rojo (se cambia a modo dirección mediante el botón DISPLAY). Los 7 ledes restantes forman una palabra binaria, con la que se indica la dirección de Profibus del Link 20E, siendo el led BF el de mayor peso y el CM el de menor. En la figura 117 se ilustra esta situación con un ejemplo.

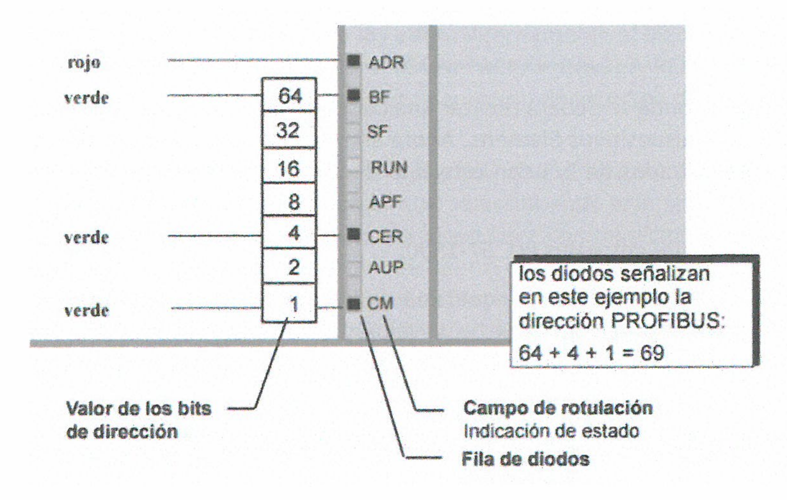


Figura 117

Para cambiar la dirección del LINK 20E, **el PLC debe estar en STOP** y se procede del siguiente modo:

- Se conmuta la indicación de DP/AS-i Link 20E apretando repetidamente el pulsador DISPLAY hasta que el diodo ADR brille con luz roja. El DP/AS-i Link 20E indica, con los siete diodos inferiores, la dirección de Profibus actualmente ajustada.
- Si se aprieta ahora el pulsador DISPLAY, el DP/AS-i Link 20E vuelve a la indicación de estado y se conserva la dirección de Profibus ajustada. Si, en cambio, se aprieta el pulsador SET, puede ajustar de nuevo la dirección de Profibus. Primero se indica, con el diodo BF parpadeante, el bit de valor más alto de la dirección de Profibus.
- Al pulsar SET, aparece este bit (diodo encendido), mientras que, si se aprieta sobre DISPLAY se anula el bit (diodo apagado). La indicación salta seguidamente al diodo SF (siguiente bit de la dirección de Profibus).
- Con la misma secuencia de operaciones descrita, se puede poner o anular consecutivamente los distintos bits de la dirección de Profibus.

Si se desea ampliar el conocimiento y características del módulo, puede visitarse la dirección indicada anteriormente y bajar el manual del DP/AS-i Link 20E, referencia 6GK1-415-2AA01.

Para configurar el mismo Link 20E pero de referencia 6GK1-415-2AA10, se procede de idéntica forma. Para mayor información sobre este módulo, visite el *website* siguiente:

https://cache.industry.siemens.com/dl/files/718/33563718/att_20354/v1/DP-ASi-Link-20E_es_2008-08_Manual_C79000-G8978-C235-01.pdf

9. Ejercicio de dos PLC S7-1500 en comunicación abierta. Enlace TCP

Enunciado

Se va a mostrar un ejemplo donde se deberá realizar una conexión abierta. Ya no se tratará de una conexión de tipo S7, solo válida para dispositivos Siemens. Ahora se realizará una conexión del tipo TCP. Todos los PLC y dispositivos descentralizados de E/S con este tipo de comunicación se podrán comunicar, sean del fabricante que sea.

Se realizará la comunicación TCP entre dos PLC S7-1500.

Las comunicaciones TCP resultan más lentas ya que, en este tipo de conexión, se prioriza el uso de las tecnologías de la información frente a la velocidad. Se ha de valorar si se desea velocidad o TI. Las comunicaciones con conexiones basadas en ISO (TCP o ISO-on-TCP) se muestran más rápidas, pero su comunicación es por MAC, no por IP.

Para establecer la comunicación, se emplearán los bloques T: TSEND_C y TRCV_C.

Material necesario

1. Dos CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Un cable de Profinet para las conexiones.

Realización

Se comenzará disponiendo sobre nuestro proyecto los dos autómatas S7-1516. Cuando estén colocados se accede a *Vista de redes*. Con la opción *Conectar en red* activada, se conecta la red entre los dos autómatas. Ahora se activa la casilla *Conexiones* y se elige el tipo de conexión *TCP*. Finalmente, se accede, de nuevo, a *Conectar en red.* En la figura 118 se representa esta situación final.

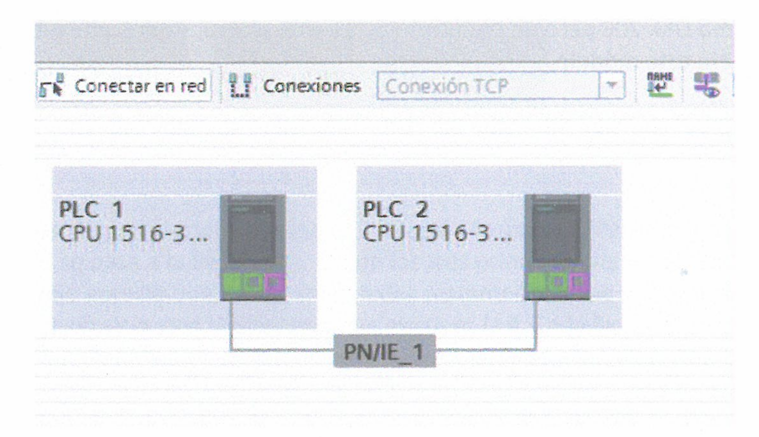


Figura 118

Una vez realizada la conexión, se efectúa la parametrización de la conexión y, para eso, se usarán bloques T, que facilitan dicha parametrización, tanto para enviar como para recibir. Se va a desarrollar un ejemplo donde uno de los PLC puede enviar y el otro, recibir.

Unidad 4 · Ejercicios prácticos de Profinet

Se utilizarán los bloques T; concretamente, el TSEND_C para enviar y el TRCV_C para recibir. Estos bloques facilitan la comunicación y, con respecto a usos de versiones anteriores, eliminan el uso de los bloques TCON, TSEND, T_DIAG, T_RESET y TDISCON, que los utiliza internamente el bloque TSEND_C. Igualmente, el bloque TRCV_C evita tener que utilizar los bloques TCON, TRCV, T_DIAG, T_RESET y TDISCON, que también los usa ese bloque internamente. De esta forma, la configuración de este tipo de conexiones se ha agilizado y facilitado.

Se accede al OB1 del PLC1. Se debe recordar que el bloque OB1 que figura por defecto no permite programar en AWL, por lo que habrá que borrarlo y añadir otro seleccionando el lenguaje AWL. Ahora se busca en el catálogo (se encuentra en la carpeta *Comunicación/Open User Communication*) el bloque para enviar, el TSEND_C, y se arrastra hasta el OB1. Al soltar el bloque dentro del OB1, se solicita un DB. Este DB será el que dispondrá de los parámetros que se necesitan para establecer, realizar y mantener la conexión. Se acepta el DB que sugiere. **Tocando** con el ratón en TSEND_C, en la parte inferior aparecerán las propiedades de la conexión, tal como se refleja en la figura 119.

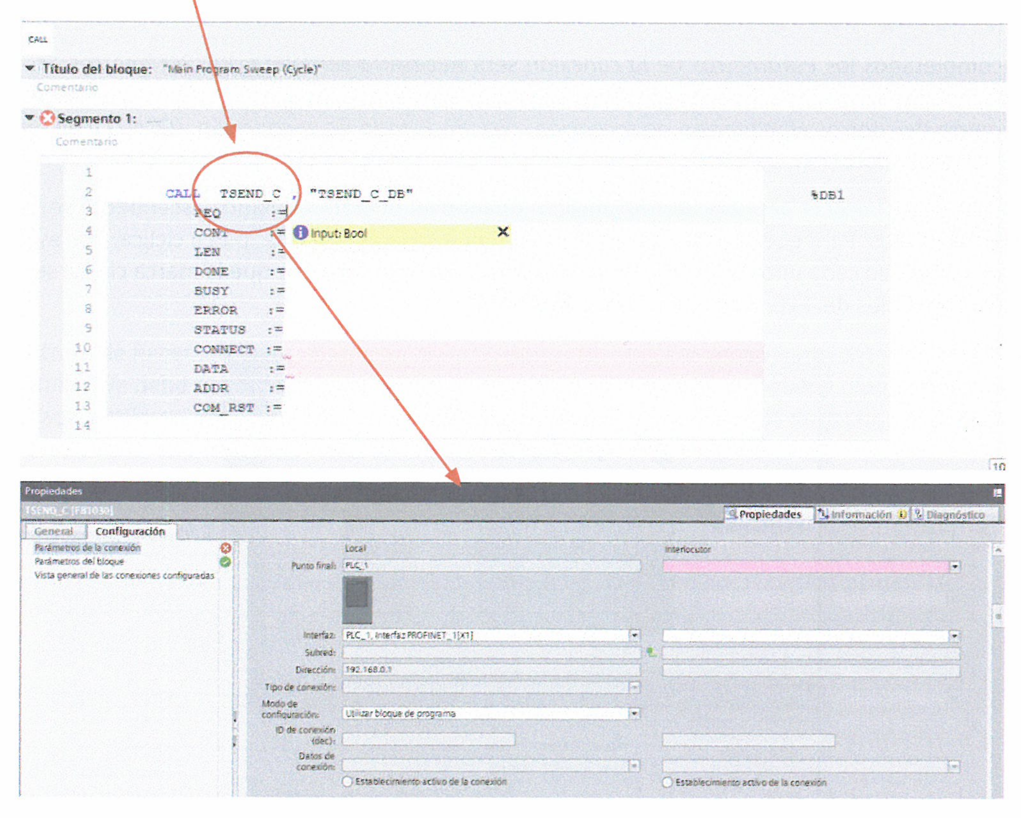


Figura 119

Se han de completar los campos necesarios para que la conexión resulte efectiva. Los pasos para introducir estos parámetros son:

1. Entre todos los PLC, se selecciona aquel al que se desea enviar datos. En este caso, solo hay uno y es el PLC2.
2. Se deben guardar los datos de conexión y, para ello, se crea un nuevo DB en la casilla *Datos de conexión*.
3. Se marca el PLC que lleva la iniciativa en la comunicación. Por defecto, es el PLC local; en este caso, el PLC1. Saldrá marcada cuando se seleccione el DB de los datos de la conexión. La opción que se debe marcar es *Establecimiento activo de la conexión* del PLC1.

Una vez realizados estos pasos, se habrán completado de forma automática los datos necesarios. En la figura 120 se muestra el resultado final de los *Parámetros de conexión*. Algunos de estos datos pueden diferir con respecto a otra aplicación, pero deben tener esa estructura.

Figura 120

Una vez completados los *Parámetros de la conexión* será necesario completar los *Parámetros del bloque*. Se pueden completar solo aquellos parámetros que sean necesarios para completar la comunicación. Estos son *Parámetro de control, Habilitación de recepción, (EN_R), Estado de la conexión, (CONT)* y en *Entradas/ Salidas* el Área de recepción, *(DATA).*

El parámetro *EN_R* inicia la petición de transmisión. Cuando se detecta un flanco ascendente se solicita la transmisión de datos. Para que sea de una manera continua se le asigna una marca cíclica. Por eso es muy importante activar en cada uno de los PLC su marca cíclica. Hay que recordar que la marca cíclica se encuentra en las propiedades del PLC, como se ve en la figura 121.

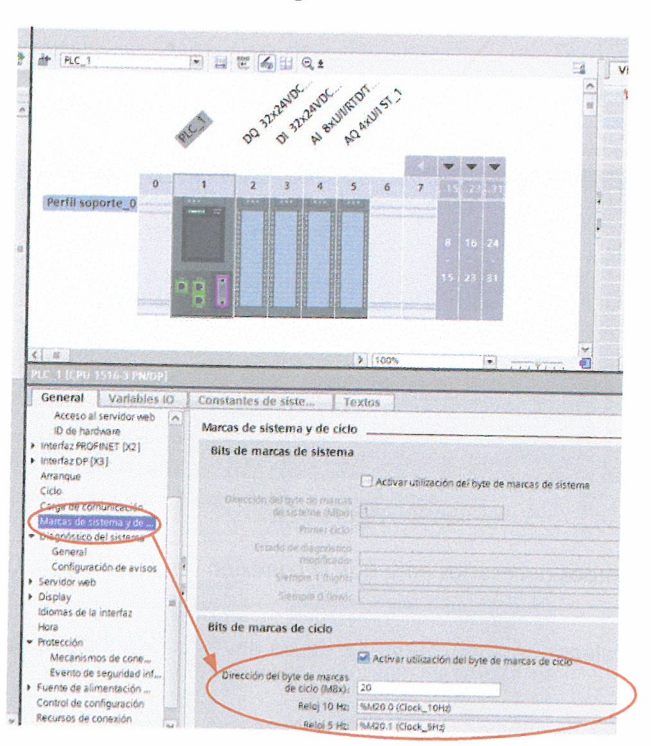


Figura 121

El parámetro *CONT* controla la conexión. Con un uno establece y mantiene la conexión y con un cero la deshace. Se colocará un UNO en este parámetro.

En *DATA* se debe colocar el puntero hacia el área de transmisión que contiene la dirección y la longitud de los datos por enviar. Para ello se indica la dirección de partida, la cantidad y el rango.

Todos estos parámetros se pueden colocar en el propio OB1 o en las propiedades de la conexión *(Parámetros de bloque)*; resulta indiferente. Colocándolo en una parte, se autocompletará la otra.

En la figura 122 se pueden ver los parámetros del bloque desde las propiedades de la conexión.

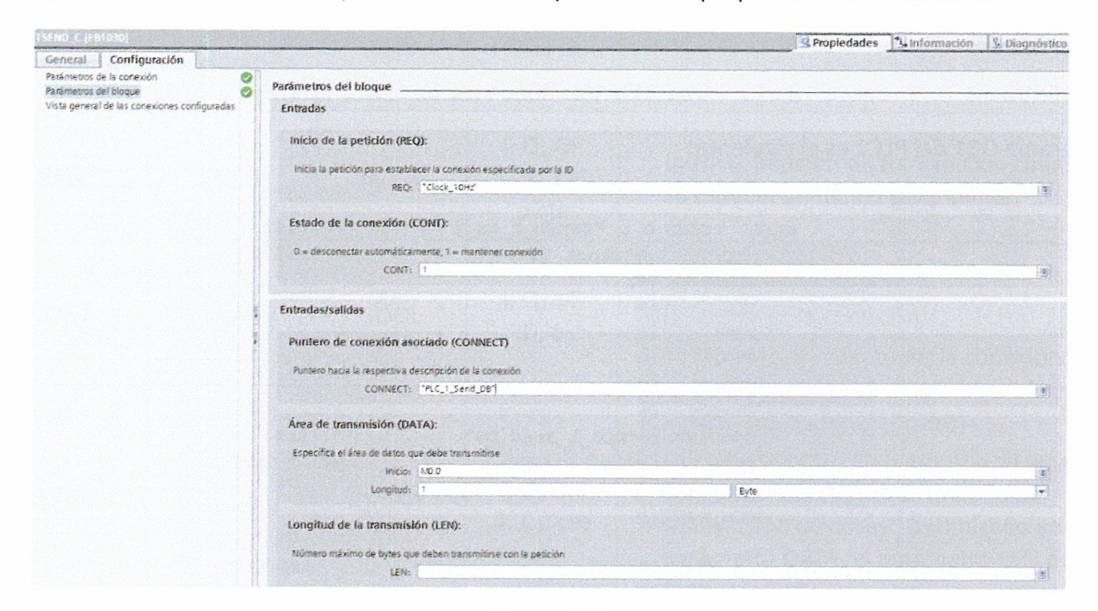


Figura 122

Ahora se trata de hacer lo mismo, pero con el PLC2, que actuará como receptor de datos. El procedimiento resulta similar. Se debe buscar el bloque *TRCV_C,* ubicado en el mismo lugar que el bloque anterior. Los parámetros son similares. En la figura 123 se refleja cómo quedan los parámetros de la conexión. Ahora, el PLC local es el PLC2 y el interlocutor, el PL1. Ya salen de forma directa de este modo.

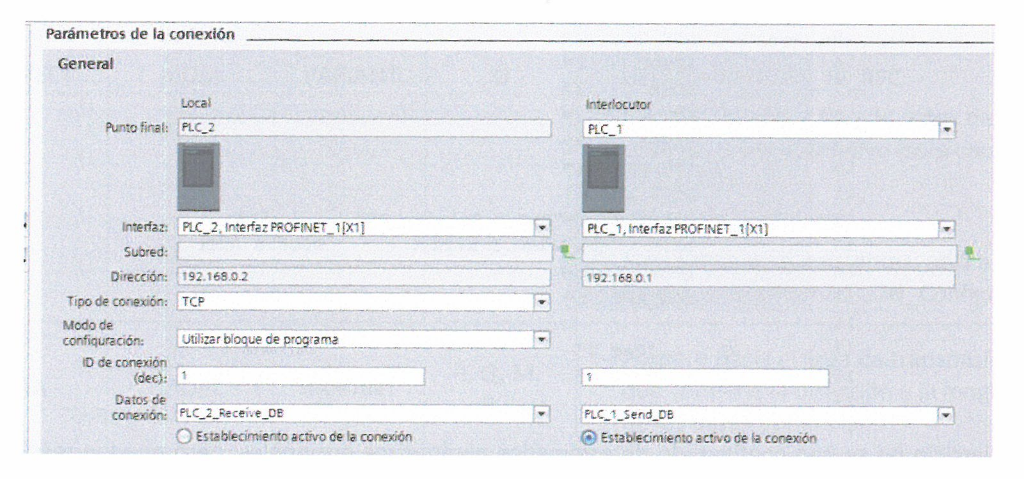


Figura 123

Enviando la configuración a cada PLC, se estará en disposición de transmitir datos entre los dos autómatas en una comunicación abierta de tipo TCP a través de Profinet. Ahora faltará comprobar que todo va bien y, para ello, se va a ejecutar un simple programa. Se deberán conocer las direcciones de la periferia de cada PLC y de las direcciones que se han configurado en los bloques de tipo T anteriores.

En este ejemplo, el PLC1 y el PLC2 presentan el siguiente direccionamiento:

PLC1:

Entradas: 0...3 Salidas: 3...0 Área DATA por donde envía: M0.0...M0.7

PLC2:

Entradas: 0...3 Salidas: 8...11 Área DATA por donde recibe: M0.0...M0.7

El programa OB1 del PLC1 es el siguiente:

```
CALL  TSEND_C , "TSEND_C_DB_1"                %DB2
    REQ      :="Clock_10Hz"                   %M20.0
    CONT     :=1                              1
    LEN      :=
    DONE     :=
    BUSY     :=
    ERROR    :=
    STATUS   :=
    CONNECT  :="PLC_1_Send_DB"                %DB3
    DATA     :=P#M0.0 BYTE 1                  P#M0.0 BYTE 1
    ADDR     :=
    COM_RST  :=

U     "Tag_1"                                 %E0.0
=     "Tag_3"                                 %M0.0
```

El programa OB1 del PLC2 es el siguiente:

```
CALL  TRCV_C , "TRCV_C_DB"                    %DB2
    EN_R      :="Clock_10Hz"                  %M20.0
    CONT      :=1                             1
    LEN       :=
    ADHOC     :=
    DONE      :=
    BUSY      :=
    ERROR     :=
    STATUS    :=
    RCVD_LEN  :=
    CONNECT   :="PLC_2_Receive_DB"            %DB1
    DATA      :=P#M0.0 BYTE 1                 P#M0.0 BYTE 1
    ADDR      :=
    COM_RST   :=

U     "Tag_1"                                 %M0.0
=     "Tag_2"                                 %A8.0
```

En este ejercicio no se han configurado determinados parámetros opcionales, pero pueden resultar muy útiles para diagnóstico cuando las cosas no van bien. A continuación, se transcribe la información que ofrece Siemens sobre el parámetro TRCV_C y TSEND_C. El texto ha sido extraído de la *Ayuda* de TIA PORTAL.

Bloque TSEND_C:

Parámetros

La tabla siguiente muestra los parámetros de la instrucción TSEND_C:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L, T, C o cte	Inicia la petición de transmisión cuando se detecta un flanco ascendente.
CONT	Input	BOOL	I, Q, M, D, L	Controla la conexión: 0: Deshacer la conexión. 1: Establecer y mantener la conexión.
LEN	Input	UDINT	I, Q, M, D, L o cte	Parámetro opcional (oculto) Número máximo de *bytes* que se envían con la petición. Si utiliza un área de transmisión de acceso optimizado en el parámetro DATA, el parámetro LEN debe tener el valor 0.
CONNECT	InOut	VARIANT	D	Puntero hacia la estructura de la descripción de la conexión Conexión programada: Con TCP o UDP, utilice el tipo de datos de sistema TCON_IP_v4. Con ISO-on-TCP, utilice el tipo de datos del sistema TCON_IP_RFC. Con ISO, utilice el tipo de datos de sistema TCON_ISOnative (solo con CP1543-1). Conexión configurada: Para conexiones existentes, utilice el tipo de datos de sistema TCON_Configured.
DATA	InOut	VARIANT	I, Q, M, D, L	Puntero hacia el área de transmisión que contiene la dirección y la longitud de los datos por enviar.
ADDR	InOut	VARIANT	D	Parámetro opcional (oculto). Puntero a la dirección del receptor.
COM_RST	InOut	BOOL	I, Q, M, D, L	Parámetro opcional (oculto). Efectúa la inicialización de la conexión: 0: Irrelevante. 1: Se inicializa la conexión existente. El parámetro COM_RST se restablece tras la evaluación mediante la instrucción TSEND_C y, por tanto, no debe interconectarse estáticamente.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
DONE	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: 0: Petición de transmisión no iniciada aún o en ejecución. 1: Petición de transmisión ejecutada sin errores. Este estado solo se muestra durante un ciclo. El parámetro de salida DONE se activa cuando finaliza correctamente un paso intermedio del procesamiento (establecer conexión, enviar, deshacer conexión) y cuando la ejecución de TSEND_C finaliza correctamente.
BUSY	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: 0: Petición de transmisión no iniciada aún o ya finalizada. 1: Petición de transmisión no finalizada aún. No es posible iniciar una nueva petición de transmisión.
ERROR	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: 0: Ningún error. 1: Se ha producido un error al establecer la conexión, al enviar los datos o al deshacer la conexión. El parámetro de salida ERROR puede activarse debido a un error en la instrucción TSEND_C o en las instrucciones de comunicación utilizadas internamente.
STATUS	Output	WORD	I, Q, M, D, L	Estado de la instrucción (véase la descripción «Parámetros ERROR y STATUS»).

Parámetros REQ, CONT y COM_RST

Con parámetro CONT, se controla el establecimiento de conexión de la instrucción TSEND_C, independientemente del parámetro REQ. El comportamiento del parámetro CONT depende, en parte, de si se utiliza una conexión programada o configurada:

- Con CONT = 0: no se envían datos (independientemente de si se utiliza una conexión programada o configurada).

- Al cambiar de CONT = 0 a 1:
  - En el caso de una conexión programada, esta se establece con TCON.
  - En el caso de una conexión configurada, esta se comprueba con T_DIAG.
- Con CONT = 1:
  - Mientras no se envíe ningún dato (REQ=0), la conexión se comprobará con T_DIAG.
  - Si las instrucciones de comunicación internas notifican que no existe ningún punto final de la conexión, esta se restablecerá automáticamente mediante TCON.
- Al cambiar de CONT = 1 a 0:
  - En el caso de una conexión programada, esta se deshace con TDISCON.
  - En el caso de una conexión configurada, esta se inicializa con T_RESET.

El parámetro COM_RST inicializa la conexión al cambiar de 0 a 1:

REQ	CONT	COM_RST	Estado de la instrucción	Descripción
Irrelevante	0	Irrelevante	No se ha ejecutado aún	Ninguna petición activa (STATUS = 7000).
Irrelevante	0	Irrelevante	Inicialización	Se está deshaciendo la conexión. La instrucción se inicializa.
Irrelevante	0 > 1	Irrelevante	Establecimiento de la conexión	La conexión se está estableciendo. Todavía no se transfieren datos.
0	1	0	Conexión establecida	La conexión se ha establecido y se vigila con la instrucción T_DIAG.
Irrelevante	1	0 > 1	Conexión establecida	La conexión se interrumpe brevemente y se inicializa con T_RESET.
0 > 1	1	0	Conexión establecida	La instrucción empieza a enviar.
Irrelevante	1	0 > 1	Enviando datos	La transferencia de datos se interrumpe. La conexión se inicializa.

Si aparece alguna conexión establecida, esta se inicializa mediante T_RESET (independientemente de si se utiliza una conexión programada o configurada).

En el caso de que no hubiera ninguna conexión establecida, la activación del parámetro no posee efecto.

Los parámetros REQ y COM_RST solo se hacen efectivos si se ha puesto CONT a 1. En la tabla de la página siguiente se muestra la relación entre los parámetros REQ, CONT y COM_RST.

Tipo de datos de sistema para conexiones configuradas

Para conexiones configuradas, utilice, en el parámetro CONNECT, la siguiente estructura para la descripción de la conexión según TCON_Configured:

Byte	Parámetro	Tipo de datos	Valor de arranque	Descripción
0 ... 1	InterfaceID	HW_ANY	-	Identificador de *hardware* de la interfaz local (rango de valores: de 0 a 65535).
2 ... 3	ID	CONN_OUC	-	Referencia a la conexión (rango de valores: de 1 a 4095). Indique la ID de la conexión existente.
4	ConnectionType	BYTE	-	Tipo de conexión. Seleccione 254 (decimal) para una conexión configurada.

Parámetros BUSY, DONE y ERROR

El estado de ejecución se controla a través de los parámetros BUSY, DONE, ERROR y STATUS. El parámetro BUSY indica el estado de ejecución. Con el parámetro DONE, puede comprobar si una petición de transmisión se ha ejecutado correctamente. El parámetro ERROR se activa si ocurren errores durante la ejecución de TSEND_C. La información de error se indica en el parámetro STATUS.

En la tabla siguiente se muestra la relación entre los parámetros BUSY, DONE y ERROR:

DONE	BUSY	ERROR	Descripción
0	0	0	La instrucción aún no se ha ejecutado (no asoma ningún flanco ascendente en el parámetro REQ).
0	1	0	La instrucción se ejecuta y establece las instrucciones de comunicación utilizadas internamente.
1	0	0	La petición de transmisión se ha ejecutado correctamente. En el parámetro STATUS se devuelve 0000. DONE = 1 solo se muestra durante un ciclo.
0	0	1	La ejecución de la instrucción o un paso intermedio del procesamiento ha finalizado con un error. Si se produce un error como consecuencia de una instrucción de comunicación utilizada internamente, se indica el error que se ha producido en primer lugar durante el procesamiento. Este estado solo se muestra durante un ciclo.

Parámetros ERROR y STATUS

ERROR	STATUS* (W#16#...)	Descripción
0	0000	Petición de transmisión ejecutada sin errores.
0	0001	Comunicación establecida.
0	0003	Comunicación desconectada.
0	7000	No se está procesando ninguna petición de transmisión. No se ha establecido ninguna conexión.
0	7001	Primera llamada al establecer una conexión.
0	7002	Segunda llamada al establecer una conexión.
0	7003	Se está deshaciendo la conexión.
0	7004	Conexión establecida y vigilada. No hay ninguna petición de transmisión activa.
0	7005	Transmisión de datos en curso.
1	80A1	• El usuario ya está utilizando la conexión o el puerto. • Error de comunicación: – La conexión indicada no se ha establecido todavía. – La conexión indicada está finalizando. No es posible transferir a través de esta conexión: – La interfaz se está reinicializando.
1	80A3	La instrucción T_DIAG subordinada ha notificado que se ha deshecho la conexión.
1	80A4	La dirección IP del punto final remoto de la conexión no es válida o bien concuerda con la dirección IP del interlocutor local.
1	80A7	Error de comunicación: se ha llamado la instrucción con COM_RST = 1 antes de finalizar la petición de transmisión.
1	80AA	Se está intentando establecer una conexión desde otro bloque con la misma ID de conexión. Repita la petición con un nuevo flanco ascendente en el parámetro REQ.
1	80B4	Al establecer una conexión pasiva (active_est = FALSE) con la variante de protocolo ISO-on-TCP (connection_type = B#16#12), se han infringido una o ambas de las condiciones siguientes: • local_tsap_id_len >= B#16#02. • local_tsap_id[1] = B#16#E0.

ERROR	STATUS* (W#16#...)	Descripción
1	80B5	Con el tipo de conexión 13 = UDP, solo se permite un establecimiento pasivo de la conexión.
1	80B6	Error de parametrización en el parámetro connection_type del bloque de datos para la descripción de la conexión.
1	80B7	Error en uno de los parámetros siguientes del bloque de datos para la descripción de la conexión: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len y next_staddr_len.
1	8085	El parámetro LEN excede el valor máximo admisible.
1	8086	El parámetro ID dentro del parámetro CONNECT está fuera del rango admisible.
1	8087	Se ha alcanzado el número máximo de conexiones. No es posible establecer más conexiones.
1	8088	El valor del parámetro LEN no es compatible con el área de recepción indicada en el parámetro DATA.
1	8089	El parámetro CONNECT no apunta a un bloque de datos.
1	8091	Se ha excedido la profundidad de anidamiento máxima.
1	809A	El parámetro CONNECT apunta a un campo que no concuerda con la longitud de la descripción de la conexión.
1	809B	InterfaceID no es válida. Es cero o no apunta a una interfaz de CPU local o un CP.
1	80C3	• Todos los recursos de conexión están ocupados. • Ya se está procesando un bloque con esta ID en otro grupo de prioridad.
1	80C4	Error de comunicación temporal: • La conexión no se puede establecer en estos momentos. • La interfaz está recibiendo nuevos parámetros o la conexión se está estableciendo. • Una instrucción TDISCON está deshaciendo la conexión configurada. • La conexión utilizada está siendo finalizada por una llamada con COM_RST = 1.
1	80C6	Error de red remoto. No es posible acceder al interlocutor remoto.
1	8722	Parámetro CONNECT: el rango de origen no es válido. El rango no existe en el DB.

ERROR	STATUS* (W#16#...)	Descripción
1	873A	Parámetro CONNECT: imposible acceder a la descripción de la conexión (p. ej., porque el DB no está disponible).
1	877F	Parámetro CONNECT: error interno.
1	8822	Parámetro DATA: el rango de origen no es válido; el rango no existe en el DB.
1	8824	Parámetro DATA: error de área en el puntero VARIANT.
1	8832	Parámetro DATA: el número de DB es demasiado alto.
1	883A	Parámetro CONNECT: imposible acceder a los datos de conexión indicados (p. ej., porque el DB no está disponible).
1	887F	Parámetro DATA: error interno (p. ej., referencia VARIANT no admisible)
1	893A	Parámetro DATA: imposible acceder al área de transmisión (p. ej., porque el DB no está disponible).

* Los códigos de error en el editor de programas se pueden representar como valores enteros o hexadecimales.

Bloque TRCV_C

Parámetros

La tabla siguiente muestra los parámetros de la instrucción TRCV_C:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
EN_R	Input	BOOL	I, Q, M, D, L, T, C o constante	Habilitación de recepción.
CONT	Input	BOOL	I, Q, M, D, L	Controla la conexión: • 0: Deshacer la conexión. • 1: Establecer la conexión y mantenerla tras recibir los datos.
LEN	Input	UDINT	I, Q, M, D, L o constante	Longitud máxima de los datos que hay que recibir. Si utiliza un área de recepción de acceso optimizado en el parámetro DATA, el parámetro LEN debe tener el valor 0.
ADHOC	Input	BOOL	I, Q, M, D, L o constante	Parámetro opcional (oculto). Utilizar el modo Ad hoc para la variante de protocolo TCP.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
CONNECT	InOut	VARIANT	D	Puntero hacia la descripción de la conexión. • Conexión programada: Con TCP o UDP, utilice la estructura TCON_IP_v4. Con ISO-on-TCP, utilice la estructura TCON_IP_RFC. Con ISO, utilice la estructura TCON_ISOnative (solo con CP1543-1). • Conexión configurada: Para conexiones existentes, utilice el tipo de datos de sistema TCON_Configured.
DATA	InOut	VARIANT	I, Q, M, D, L	Puntero hacia el área de recepción.
ADDR	InOut	VARIANT	D	Parámetro opcional (oculto). Puntero hacia la dirección del emisor con el tipo de conexión UDP.
COM_RST	InOut	BOOL	I, Q, M, D, L	Parámetro opcional (oculto). Efectúa la inicialización de la conexión: 0: Irrelevante. 1: Se inicializa la conexión existente. El parámetro COM_RST se restablece tras la evaluación mediante la instrucción TRCV_C y, por tanto, no debe interconectarse estáticamente.
DONE	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: 0: Recepción no iniciada aún o en ejecución. 1: Recepción ejecutada sin errores. Este estado solo se muestra durante un ciclo. El parámetro de salida DONE se activa cuando finaliza correctamente un paso intermedio del procesamiento (establecer conexión, recepción, deshacer conexión) y cuando la ejecución de TRCV_C finaliza correctamente.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
BUSY	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: • 0: Recepción no iniciada aún o ya finalizada. • 1: Recepción no finalizada todavía. No es posible iniciar una nueva petición de transmisión.
ERROR	Output	BOOL	I, Q, M, D, L	Parámetro de estado con los valores siguientes: • 0: Ningún error. • 1: Se ha producido un error al establecer la conexión, al recibir los datos o al deshacer la conexión. El parámetro de salida ERROR puede activarse debido a un error en la instrucción TRCV_C o en las instrucciones de comunicación utilizadas internamente.
STATUS	Output	WORD	I, Q, M, D, L	Estado de la instrucción.
RCVD_LEN	Output	UDINT	I, Q, M, D, L	Cantidad de datos (en *bytes*) recibida realmente.

Parámetros EN_R, CONT y COM_RST

El parámetro CONT controla el establecimiento de conexión de la instrucción TRCV_C, independientemente del parámetro EN_R. El comportamiento del parámetro CONT depende, en parte, de si se utiliza una conexión programada o configurada:

- Con CONT = 0: no se reciben datos (independientemente de si se emplea una conexión programada o configurada).
- Al cambiar de CONT=0 a 1:
  - En el caso de una conexión programada, esta se establece con TCON.
  - En el caso de una conexión configurada, esta se comprueba con T_DIAG.
- Con CONT = 1:
  - Mientras no se reciba ningún dato (EN_R=0), se comprueba la conexión con T_DIAG.
  - Si las instrucciones de comunicación internas notifican que no existe ningún punto final de la conexión, esta se restablecerá automáticamente mediante TCON.
- Al cambiar de CONT=1 a 0:
  - En el caso de una conexión programada, esta se deshace con TDISCON.
  - En el caso de una conexión configurada, esta se inicializa con T_RESET.

El parámetro COM_RST inicializa la conexión al cambiar de 0 a 1:

- Si existe una conexión establecida, esta se inicializa mediante T_RESET (independientemente de si se utiliza una conexión programada o configurada).
- En el caso de no haber ninguna conexión establecida, la activación del parámetro no posee efecto.

Los parámetros EN_R y COM_RST solo se hacen efectivos si se ha puesto CONT a 1. La tabla siguiente muestra la relación entre los parámetros EN_R, CONT y COM_RST:

EN_R	CONT	COM_RST	Estado de la instrucción	Descripción
Irrelevante	0	Irrelevante	No se ha ejecutado aún	Ninguna petición activa (STATUS = 7000).
Irrelevante	0	Irrelevante	Inicialización	Se está deshaciendo la conexión. La instrucción se inicializa.
Irrelevante	0 > 1	Irrelevante	Establecimiento de la conexión	La conexión se está estableciendo. Todavía no se transfieren datos.
0	1	0	Conexión establecida	La conexión se ha establecido y se vigila con la instrucción T_DIAG.
Irrelevante	1	0 > 1	Conexión establecida	La conexión se interrumpe brevemente y se inicializa con T_RESET.
0 > 1	1	0	Conexión establecida	La instrucción empieza a recibir.
Irrelevante	1	0 > 1	Recibiendo datos	La transferencia de datos se interrumpe. La conexión se inicializa.

Tipo de datos de sistema para conexiones configuradas

Para conexiones configuradas, utilice, en el parámetro CONNECT, la siguiente estructura para la descripción de la conexión según TCON_Configured:

Byte	Parámetro	Tipo de datos	Valor de arranque	Descripción
0 ... 1	InterfaceID	HW_ANY	-	Identificador de *hardware* de la interfaz local (rango de valores: de 0 a 65535).
2 ... 3	ID	CONN_OUC	-	Referencia a la conexión (rango de valores: de 1 a 4095). Indique la ID de la conexión existente.
4	ConnectionType	BYTE	-	Tipo de conexión Seleccione 254 (decimal) para una conexión configurada.

Parámetros BUSY, DONE y ERROR

El estado de ejecución se controla a través de los parámetros BUSY, DONE, ERROR y STATUS. El parámetro BUSY indica el estado de ejecución. El parámetro DONE permite comprobar si una petición se ha ejecutado correctamente. El parámetro ERROR se activa si ocurren errores durante la ejecución de TRCV_C. La información de error se indica en el parámetro STATUS.

En la tabla siguiente se muestra la relación entre los parámetros BUSY, DONE y ERROR:

DONE	BUSY	ERROR	Descripción
0	0	0	La instrucción aún no se ha ejecutado (no asoma ningún flanco ascendente en el parámetro EN_R).
0	1	0	La instrucción se ejecuta y llama a las instrucciones de comunicación utilizadas internamente.
1	0	0	La recepción se ha ejecutado correctamente. En el parámetro STATUS se devuelve 0000. DONE = 1 solo se muestra durante un ciclo.
0	0	1	La ejecución de la instrucción o un paso intermedio del procesamiento ha finalizado con un error. Si se produce un error como consecuencia de una instrucción de comunicación utilizada internamente, se indica el error que se ha producido en primer lugar durante el procesamiento. Este estado solo se muestra durante un ciclo.

Parámetros ERROR y STATUS

ERROR	STATUS (W#16#...)	Descripción
0	0000	Petición de recepción ejecutada sin errores.
0	0001	Comunicación establecida.
0	0003	Comunicación desconectada.
0	7000	No se está procesando ninguna petición.
0	7001	Primera llamada al establecer una conexión.
0	7002	Segunda llamada al establecer una conexión.
0	7003	Se está deshaciendo la conexión.
0	7004	Conexión establecida y vigilada. No se está procesando ninguna petición de recepción.

ERROR	STATUS (W#16#...)	Descripción
0	7006	Se están recibiendo datos.
1	8085	• El parámetro LEN excede el valor máximo admisible. • El valor del parámetro LEN o DATA ha sido modificado después de la primera llamada.
1	8086	El parámetro ID se encuentra fuera del rango admisible.
1	8087	Se ha alcanzado el número máximo de conexiones. No es posible establecer más conexiones.
1	8088	El valor del parámetro LEN no resulta compatible con el área de recepción indicada en el parámetro DATA.
1	8089	El parámetro CONNECT no apunta a un bloque de datos.
1	8091	Se ha excedido la profundidad de anidamiento máxima.
1	809A	El parámetro CONNECT apunta a un campo que no concuerda con la longitud de la descripción de la conexión.
1	809B	InterfaceID no es válida. Es cero o no apunta a una interfaz de CPU local o un CP.
1	80A1	• El usuario ya está utilizando la conexión o el puerto. • Error de comunicación: – La conexión indicada no se ha establecido todavía. – La conexión indicada se está finalizando. No es posible transferir a través de esta conexión: – La interfaz se está reinicializando.
1	80A3	La instrucción T_DIAG subordinada ha notificado que se ha deshecho la conexión.
1	80A4	La dirección IP del punto final remoto de la conexión no es válida o bien concuerda con la dirección IP del interlocutor local.
1	80A7	Error de comunicación: se ha llamado la instrucción con COM_RST=1 antes de finalizar la petición de transmisión.
1	80AA	Se está intentando establecer una conexión desde otro bloque con la misma ID de conexión. Repita la petición con un nuevo flanco ascendente en el parámetro REQ.
1	80B4	Al establecer una conexión pasiva (active_est=FALSE) con la variante de protocolo ISO-on-TCP (connection_type=B#16#12), se han infringido una o ambas de las condiciones siguientes: "local_tsap_id_len >= B#16#02" y/o "local_tsap_id[1]=B#16#E0".
1	80B5	Con el tipo de conexión 13 = UDP solo se permite un establecimiento pasivo de la conexión.

ERROR	STATUS (W#16#...)	Descripción
1	80B6	Error de parametrización en el parámetro connection_type del bloque de datos para la descripción de la conexión.
1	80B7	Error en uno de los parámetros siguientes del bloque de datos para la descripción de la conexión: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len y next_staddr_len.
1	80C3	• Todos los recursos de conexión están ocupados. • Ya se está procesando un bloque con esta ID en otro grupo de prioridad.
1	80C4	Error de comunicación temporal: • La conexión no se puede establecer en estos momentos. • La interfaz está recibiendo nuevos parámetros o la conexión se está estableciendo. • Una instrucción TDISCON está deshaciendo la conexión configurada. • La conexión utilizada está siendo finalizada por una llamada con COM_RST=1.
1	80C6	No es posible acceder al interlocutor remoto (error de red).
1	8722	Error en el parámetro CONNECT: rango de origen no válido (el rango no está declarado en el bloque de datos).
1	873A	Error en el parámetro CONNECT: el acceso a la descripción de la conexión no es posible (no existe acceso al bloque de datos).
1	877F	Error en el parámetro CONNECT: error interno.
1	8922	Parámetro DATA: el rango de destino no es válido; el rango no existe en el DB.
1	8924	Parámetro DATA: error de área en el puntero VARIANT.
1	8932	Parámetro DATA: el número de DB es demasiado alto.
1	893A	Parámetro CONNECT: imposible acceder a los datos de conexión indicados (p. ej., porque el DB no está disponible).
1	897F	Parámetro DATA: error interno (p. ej., referencia VARIANT no admisible).
1	8A3A	Parámetro DATA: imposible acceder al área de datos, p. ej., porque el bloque de datos no existe.

Ejercicio propuesto

En el anterior ejemplo no se ha configurado la opción para que el PLC1 reciba del PLC2 ni que el PLC2 pueda enviar al PLC1. Se propone incluir las modificaciones necesarias para que se produzca la comunicación en ese sentido.

10. Ejercicio de comunicación de PLC S7-1500, S7-1200, S7-300 y periferia

Enunciado

En el siguiente ejercicio se van a comunicar tres PLC, un S7-1500, un S7-300 y un S7-1200; concretamente, el PLC S7 1516 3PN/DP, el S7-315 2PN/DP y el S7-1214C AC/DC/RLY. También se dispone de una estación de E/S descentralizada.

Material necesario

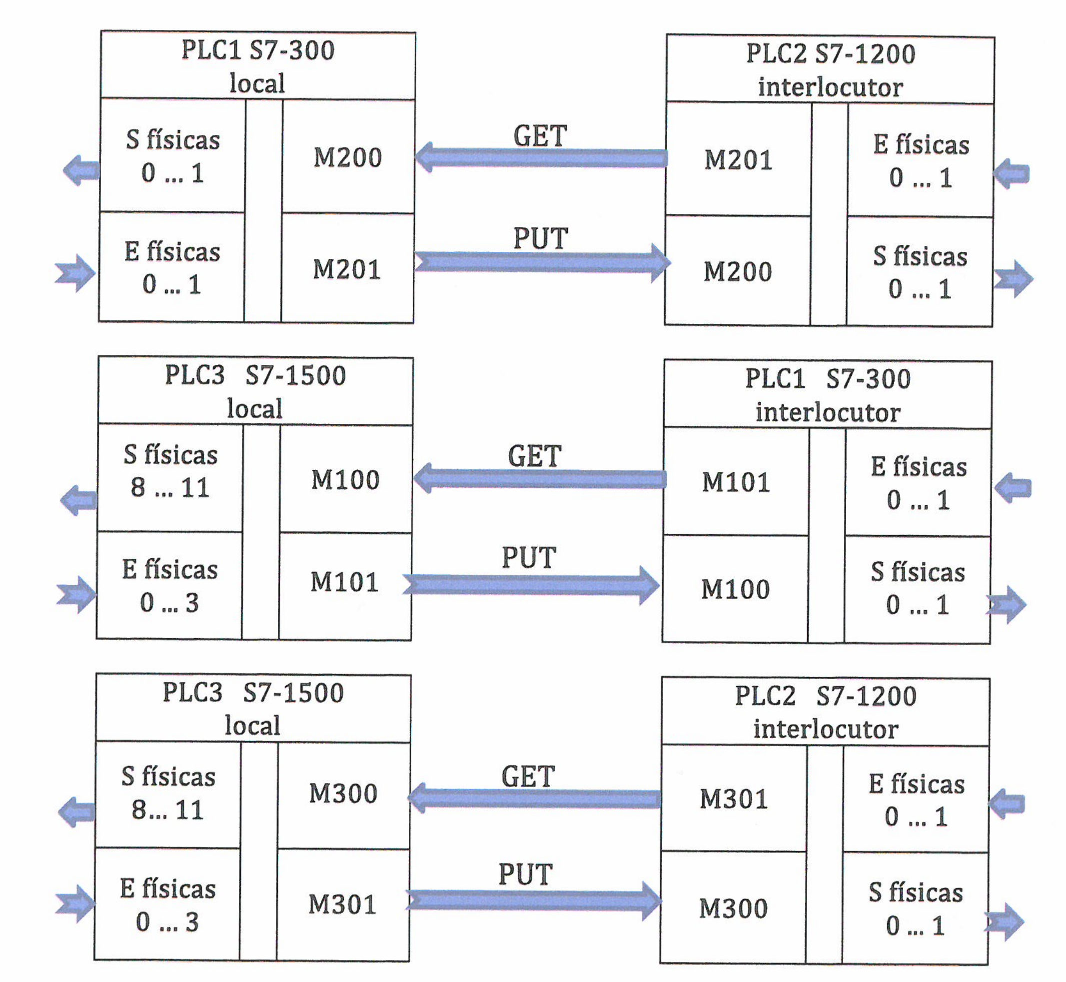
1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una CPU 315 2PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación SITOP de 5 A.
   - Un módulo de 16 entradas digitales a 24 V (DC).
   - Un módulo de 16 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas y 2 salidas analógicas.
3. Una CPU 1214C con los siguientes módulos compactos:
   - 14 entradas digitales.
   - 10 salidas digitales.
   - 1 salida analógica.
4. Una ET 200S IM 151-3PN con 5 módulos de entradas digitales y 5 módulos de salidas digitales.
5. Un cable de Profinet para las conexiones.

Realización

Anteriormente se han descrito ejemplos donde se realizan conexiones de tipo S7 utilizando los bloques GET/PUT. Revise el ejercicio 5 para recordar el procedimiento, ya que, en este caso, es lo mismo, pero con más conexiones.

Lo primero que se debe hacer es configurar de forma individual cada uno de los PLC y cargar dicha configuración a cada uno de ellos individualmente.

Una vez realizada la carga de cada PLC, se establece cómo se comunican; es decir, se establecen las conexiones entre ellos: quién se va a comunicar con quién. Como todos los dispositivos son de Siemens, se va a utilizar conexión S7. En la siguiente tabla se muestran las direcciones de E/S físicas que tiene cada PLC, así como las direcciones de las marcas que se emplearán en la configuración de cada bloque GET/PUT.



Antes de seguir, resulta necesario que el PLC 1200 disponga de permiso de utilización para este tipo de conexión, mediante GET/PUT. Por defecto, viene deshabilitada. Para ello, se ha de ir a las propiedades del 1200 y, en *Protección,* marcar la casilla correspondiente, tal como aparece en la figura 124. **Lo mismo ocurre en el PLC 1500.**

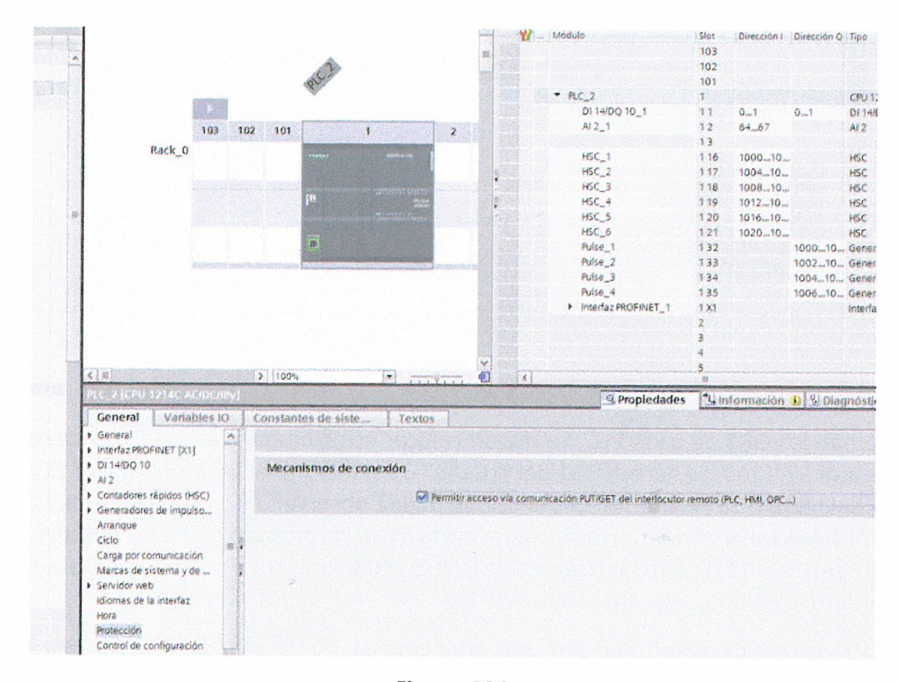


Figura 124

Se establece la red entre los tres PLC y la ET 200. Se debe hacer una conexión del tipo S7. También se ha de configurar una marca de ciclo para las funciones GET/PUT en los PLC S7-300 y S7-1500. El S7-1200 no la necesita, ya que nunca es local y, por lo tanto, no dispone de los bloques GET/PUT. En la figura 125 se indica el resultado.

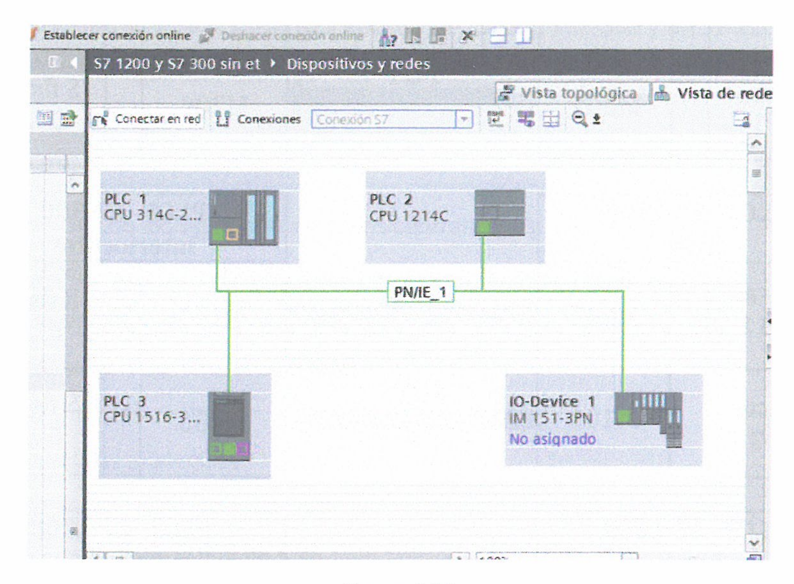


Figura 125

Se observa que la ET 200S no se encuentra asignada a ningún PLC. Solo un PLC podrá acceder directamente a dicha ET. En la figura 126 se ve ya asignada la ET; en este caso, al PLC3, que es el PLC 1516.

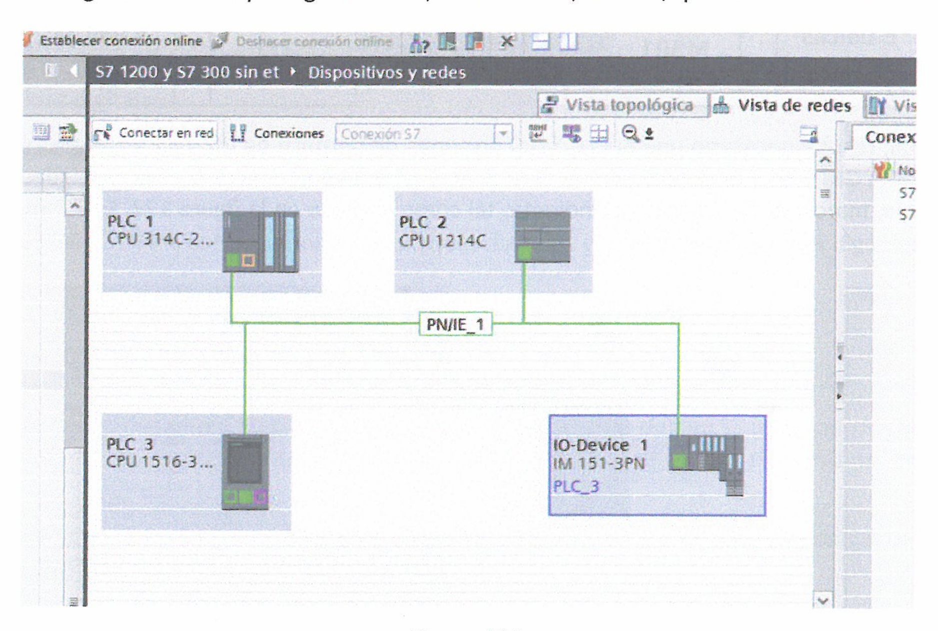


Figura 126

Ahora se programan los OB1 de cada PLC. Se deben colocar estos bloques en los PLC locales. El S7-300 es local del interlocutor S7-1200. Se abre el OB1 del PLC1 (S7-300) y se sitúan los bloques GET y PUT. Ahora, con el ratón encima del texto GET y pulsando dos veces, se puede observar, en la parte inferior, las propiedades de la conexión. Al seleccionar el interlocutor, aparecerá la ID de la conexión de forma automática. Lo mismo se hará en el OB1 del PLC3 1516, que también es local de los interlocutores PLC1 y PLC2. Deberá completarse el resto de parámetros, como se hizo en el ejercicio 3.

En la figura 127 se indica cómo quedan las propiedades de la conexión entre el PLC1 (S7-300) y el PLC2 (S7-1200).

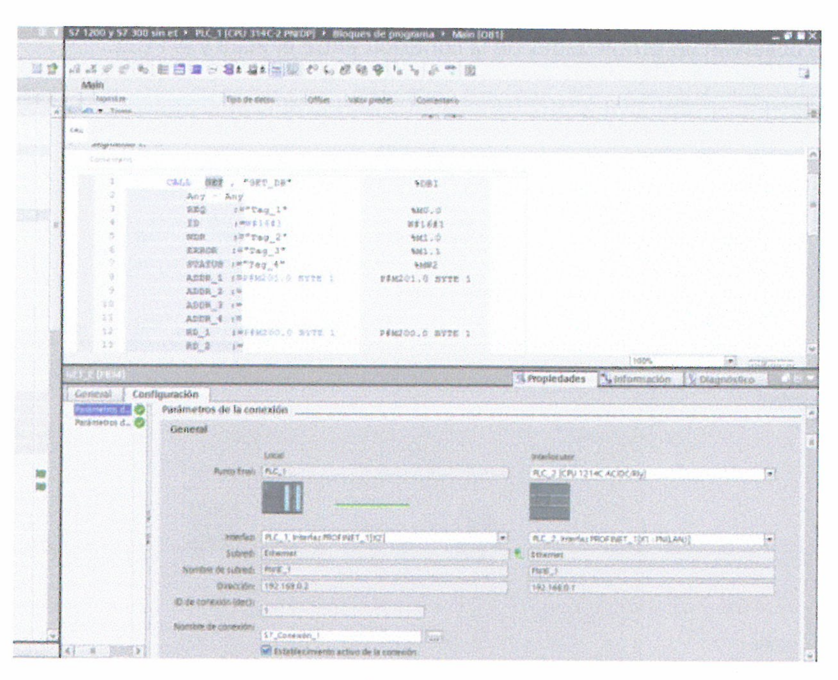


Figura 127

En la figura 128 se pueden ver las propiedades de la conexión entre el PLC3 y el PLC1.

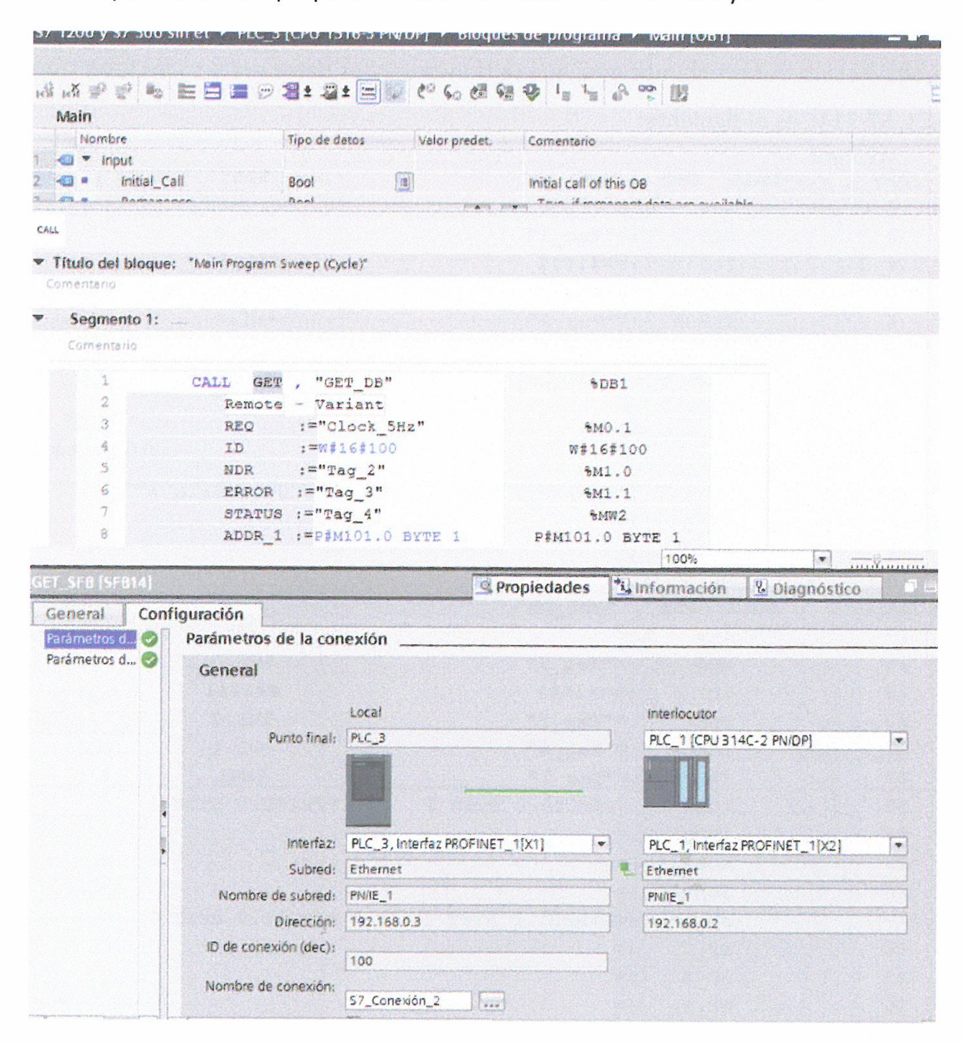


Figura 128

En la figura 129 se observan las propiedades de la conexión entre el PLC3 y el PLC2.

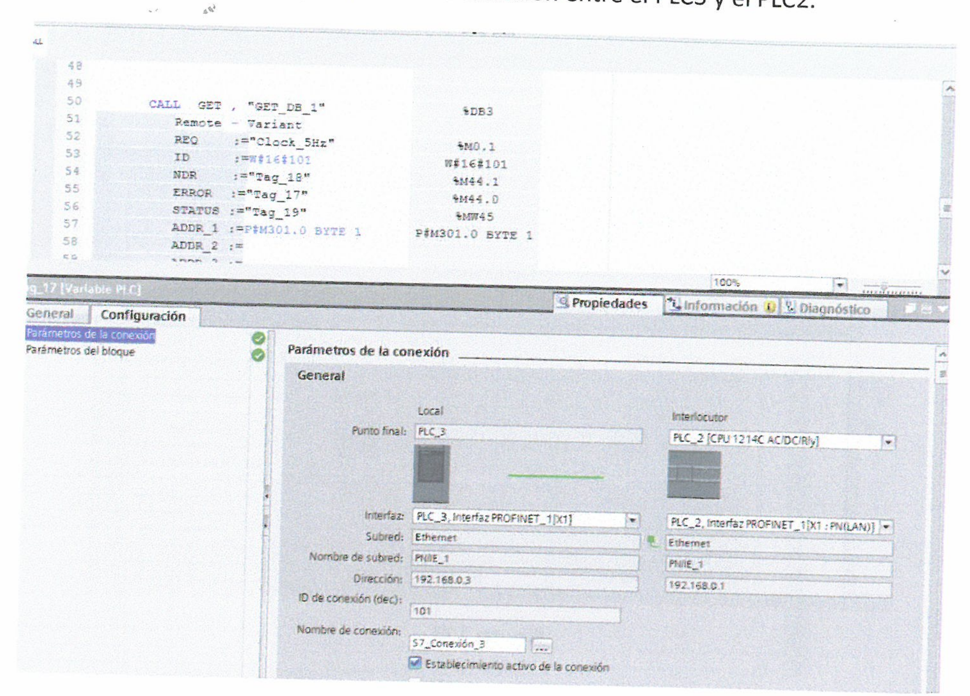


Figura 129

Ahora solo queda ejecutar los programas para que se transmitan los datos entre todos los dispositivos.

El OB1 del PLC1, S7-314C es el siguiente:

```
CALL  GET , "GET_DB"                 %DB1
      Any - Any
      REQ    :="Tag_1"               %M0.0
      ID     :=W#16#1                W#16#1
      NDR    :="Tag_2"               %M1.0
      ERROR  :="Tag_3"               %M1.1
      STATUS :="Tag_4"               %MW2
      ADDR_1 :=P#M201.0 BYTE 1       P#M201.0 BYTE 1
      ADDR_2 :=
      ADDR_3 :=
      ADDR_4 :=
      RD_1   :=P#M200.0 BYTE 1       P#M200.0 BYTE 1
      RD_2   :=
      RD_3   :=
      RD_4   :=
CALL  PUT , "PUT_DB"                 %DB2
      Any - Any
      REQ    :="Tag_1"               %M0.0
      ID     :=W#16#1                W#16#1
      DONE   :="Tag_5"               %M3.0
      ERROR  :="Tag_6"               %M3.1
      STATUS :="Tag_7"               %MW4
      ADDR_1 :=P#M200.0 BYTE 1       P#M200.0 BYTE 1
      ADDR_2 :=
      ADDR_3 :=
      ADDR_4 :=
      SD_1   :=P#M201.0 BYTE 1       P#M201.0 BYTE 1
      SD_2   :=
      SD_3   :=
      SD_4   :=
```



```
U  "Entrada_E0.0"         %E0.0
=  "Envía_al_1200"        %M201.0

U  "Recibe_del_1200"      %M200.0
=  "Salida_A0.0"          %A0.0

U  "Recibe_del_1500"      %M100.0
=  "Salida_A0.1"          %A0.1

U  "Entrada_E0.1"         %E0.1
=  "Envía_al_1500"        %M101.0

U  "Entrada_E0.2"         %E0.2
=  "Envía_al_1500_2"      %M101.2
```

A continuación, se ilustra el programa del PLC2 (S7-1200):

Segmento 1:

Comentario

%M200.0 "Recibe_del_300" —| |— —( )— %A0.0 "Salida_A0.0"

Segmento 2:

Comentario

%E0.0 "Entrada_E0.0" —| |— —( )— %M301.0 "Envía_al_1500"

Segmento 3:

Comentario

%E0.0 "Entrada_E0.0" —| |— —( )— %M201.0 "Envía_al_300"

Segmento 4:

Comentario

%M300.0 "Recibe_del_1500" —| |— —( )— %A0.1 "Salida_A0.1"

Por último, el programa del autómata PLC3 (S7-1500) es el siguiente:

```
CALL  GET , "GET_DB"                          %DB1
  Remote  - Variant
  REQ     :="Clock_5Hz"                       %M0.1
  ID      :=W#16#100                          W#16#100
  NDR     :="Tag_2"                           %M1.0
  ERROR   :="Tag_3"                           %M1.1
  STATUS  :="Tag_4"                           %MW2
  ADDR_1  :=P#M101.0 BYTE 1                   P#M101.0 BYTE 1
  ADDR_2  :=
  ADDR_3  :=
  ADDR_4  :=
  RD_1    :=P#M100.0 BYTE 1                   P#M100.0 BYTE 1
  RD_2    :=
  RD_3    :=
  RD_4    :=
CALL  PUT , "PUT_DB"                          %DB2
  Remote  - Variant
  REQ     :="Clock_5Hz"                       %M0.1
  ID      :=w#16#100                          w#16#100
  DONE    :="Tag_5"                           %M4.0
  ERROR   :="Tag_6"                           %M4.1
  STATUS  :="Tag_7"                           %MW5
  ADDR_1  :=P#M100.0 BYTE 1                   P#M100.0 BYTE 1
  ADDR_2  :=
  ADDR_3  :=
  ADDR_4  :=
  SD_1    :=P#M101.0 BYTE 1                   P#M101.0 BYTE 1
  SD_2    :=
  SD_3    :=
  SD_4    :=
```


11. Ejercicio de integración de redes Profibus, Profinet y AS-i

Enunciado

Se va a realizar un ejercicio donde convivan las tres comunicaciones más empleadas actualmente: Profibus, Profinet y AS-i. Para integrarlas, se utilizará un PLC 1500 que tiene integrado las interfaces de Profibus y Profinet. A ese sistema se le añadirá la red de AS-i mediante el Link 20E.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una ET 200S IM 151-3PN con 5 módulos de entradas digitales y 5 módulos de salidas digitales. Interfaz de Profinet.
3. Una ET 200M IM 153-1DP con un módulo de entradas y salidas digitales de 16.
4. Un Link DP/AS-i Link 20E de referencia 6GK1415-2AA01 y diversos esclavos E/S de AS-i.
4. Un cable de Profinet y Profibus para las conexiones.

Realización

Como ya se ha dicho, el objetivo del ejercicio es la convivencia de las tres redes, de forma que pueda haber comunicación entre todos los módulos de E/S que pertenecen a redes distintas. Es un ejercicio muy real ya que las modificaciones y actualizaciones que se realizan en la industria no siempre se pueden ejecutar todas de una sola vez. Por ello, resulta muy útil que convivan diferentes sistemas de comunicación.

Como siempre, lo primero que se debe hacer es configurar el autómata y colocar las dos ET que se van a conectar, una de Profibus y otra de Profinet.

Una vez dispuestos todos los dispositivos, se establecerá la conexión entre ellos. La ET de Profibus se conecta a la interfaz de Profibus del PLC y la ET de Profinet a la interfaz de Profinet, tal como se aprecia en la figura 130.

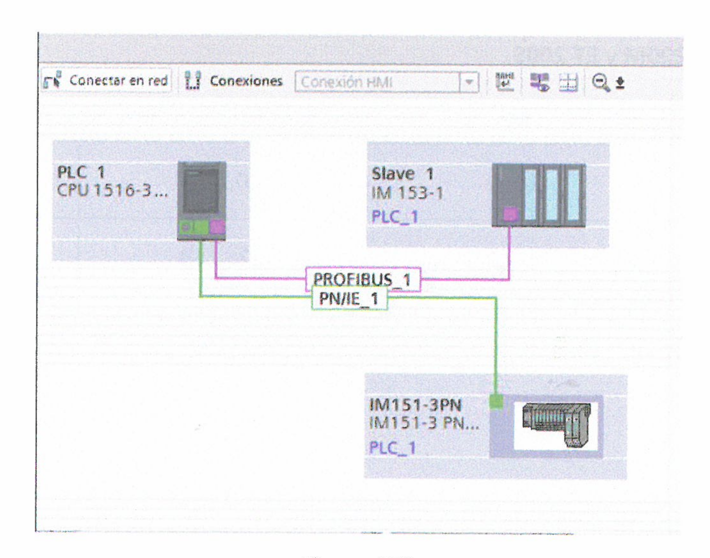


Figura 130

Falta por añadir el Link DP/AS-i 20E. Recuérdese cómo se hizo en el ejercicio número 8. Los pasos que se deben seguir para completar su configuración son los siguientes:

1. Se busca en el catálogo el Link E20 de referencia adecuada y, si no estuviera, se baja su GSD.
2. Se coloca en *Vista de redes*.
3. Se añade en la parte derecha, el módulo de reserva de memoria, como indican las figuras 102, 103 y 104.
4. Se asigna el Link 20E al PLC.
5. Se inscribe en el Link 20E la dirección real que tiene y se comprueba la dirección, como se establece en el ejercicio indicado.
6. Se verifica la primera dirección del Link 20E y, con ello, se entra en la tabla para conocer la dirección de cada uno de los esclavos siguiendo el procedimiento indicado en el ejercicio 6.

Una vez realizado esto, se está en disposición de enviar toda la configuración al PLC. Antes hay que asegurarse de que la ET 200S de Profinet tiene en el proyecto el mismo nombre que en la realidad. A tal fin, se recuerda que se ha de estar *online* y que se debe comprobar si corresponde el nombre real con el del proyecto. Si no fuese así, se sitúa en *Online y diagnóstico* y, desde, *Funciones* se modifica el nombre. En la figura 131 se visualiza el sistema conectado correctamente.

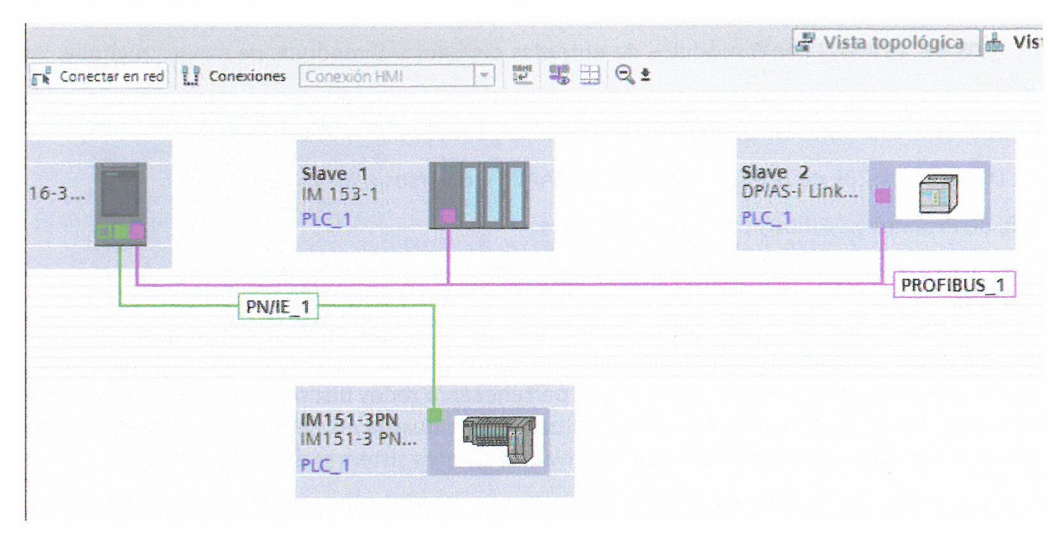


Figura 131

Para poder llevar a cabo el programa, hay que conocer todas las direcciones de los diferentes participantes. En la siguiente tabla se indican dichas direcciones para este ejemplo. Evidentemente, cualquier otra configuración, con otras ET y otros esclavos de AS-i, mostrará direcciones diferentes.

Direcciones del PLC, ET 200M y ET 200S

	PLC	ET 200M (Profibus)	ET 200S (Profinet)
Entradas	0... 3	20... 21	30... 34
Salidas	0... 3	12 ... 13	18... 22

Direcciones de los esclavos de AS-i

	Primera dirección (m)	m + valor que sale en la tabla de AS-i	Dirección esclavo de AS-i	*Byte*
Columna de señalización (3 salidas)	40	m + 6	46	46. 4-46. 7
Contactores (2 salidas)	40	m + 6	46	46. 0-46. 3
Detector (1 entrada)	40	m + 2	42	42. 4-42. 3
Pulsadores (2 entradas)	40	m + 7	47	47. 4-47. 7

Para establecer la comunicación entre todos, tan solo se ha de llamar a la dirección de forma directa. No es necesaria ninguna conexión (enlace), ya que la comunicación se muestra directa.

Con el siguiente programa, se entabla comunicación entre los diferentes módulos; así, se puede observar lo fácil que es integrar diferentes tipos de protocolos de comunicación:

```
// Activación de 8 salidas de las ET-s de Profibus y Profinet
// mediante la entrada EB0 del PLC.
        L       "Entrada_B0_PLC"                        %EB0
        T       "Salida_B12_ET_Profibus"                %AB12
        T       "Salida_B20_ET_Profinet"                %AB20

// Una entrada del PLC activa los esclavos de AS-i
        U       "Entrada_E1.0_PLC"                      %E1.0
        =       "Salida1_columna_ASi"                   %A46.4
        =       "Salida2_columna_ASi"                   %A46.5
        =       "Salida3_columna_ASi"                   %A46.6
        =       "Salida1_contactor_ASi"                 %A46.0
        =       "Salida2_contactor_ASi"                 %A46.1

// Una entrada de AS-i (detector inductivo) activa una
// salida del PLC y las salidas de las ET-s Profibus y Profinet


        U       "Entrada_detector_ASi"                  %E42.4
        =       "Salida_A1.0_ASi"                       %A1.0
        =       "Salida_A13.0_ET_Profibus"              %A13.0
        =       "Salida_A18.0_ET_Profinet"              %A18.0
```

12. Ejercicio de acoplamiento de dos redes de distintas subredes

Enunciado

Unir dos redes de diferentes grupos, es decir, dos redes que no pertenecen al mismo entorno, resulta útil por muchas razones; una de ellas por seguridad, pues sirve para separar una red industrial de otra ofimática dentro de una misma empresa. Otra razón estriba en la necesidad de unir dos plataformas que disponen de grupos diferentes sin tener que modificar las configuraciones de uno de ellos. Las dos redes que se deben acoplar han de ser Ethernet.

Material necesario

1. Dos CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Un acoplador PN/PN.
3. Un cable de Profinet para las conexiones.

Realización

Se describe cómo configurar un PN/PN Coupler; concretamente, el de referencia 6ES7 158-3AD00-0XA0.

En primer lugar se disponen los dos PLC en el proyecto. Lo siguiente será colocar el acoplador. Este elemento va a unir dos redes de Profinet de distinto grupo. Se podrá intercambiar información entre las dos redes. El acoplador se compone de dos lados, X1 y X2, uno para cada red que se vaya a enlazar. En la figura 132 se muestra el acoplador.

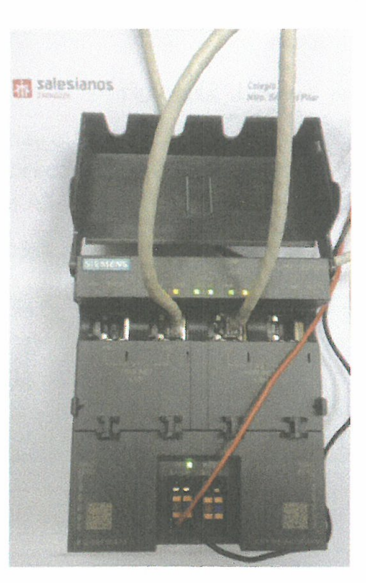


Figura 132

Desde *Vista de redes,* habrá que buscar en el catálogo el acoplador. Una vez encontrado, con la referencia exacta se colocan las dos partes del acoplador: X1 y X2. En la figura 133 se observa el lugar del catálogo donde se encuentra el acoplador.

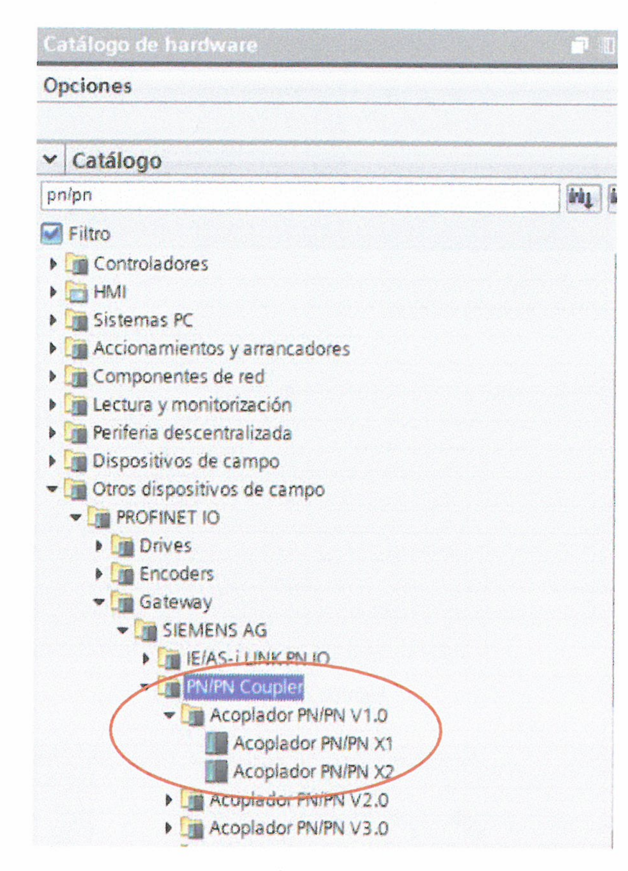


Figura 133

Tras colocar las dos partes del acoplador y los dos PLC, el proyecto quedará como se aprecia en la figura 134.

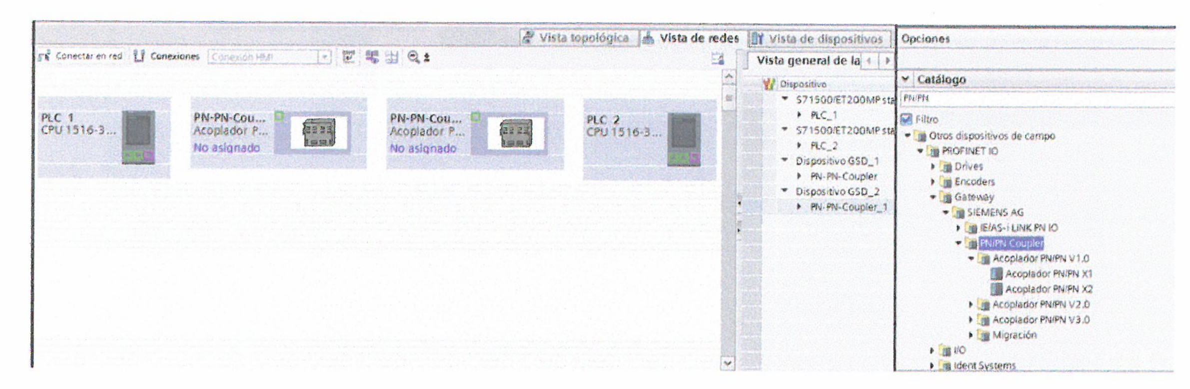


Figura 134

Ahora se deben aportar direcciones de IP a cada dispositivo. El acoplador cuenta con dos direcciones de IP: una para el lado X1 y otra para el lado X2. Cada PLC se acoplará a un lado de dicho acoplador. En este caso, el PLC1 se pondrá del lado X1 y el PLC2, del lado X2.

TIA PORTAL habrá dado a cada PLC direcciones dentro de un mismo grupo. Se debe cambiar un PLC y su correspondiente parte del acoplador para que presenten direcciones de distinto grupo. En este ejemplo, se van a dar direcciones de dos grupos: 192.168.0.x y el 192.168.1.x. Se adscribe la dirección al PLC y luego, cuando se desarrolle la red con el acoplador, se colocará una dirección acorde con la del PLC. En la figura 135 se indica esta situación en el PLC1 y en la figura 136, en el PLC2.

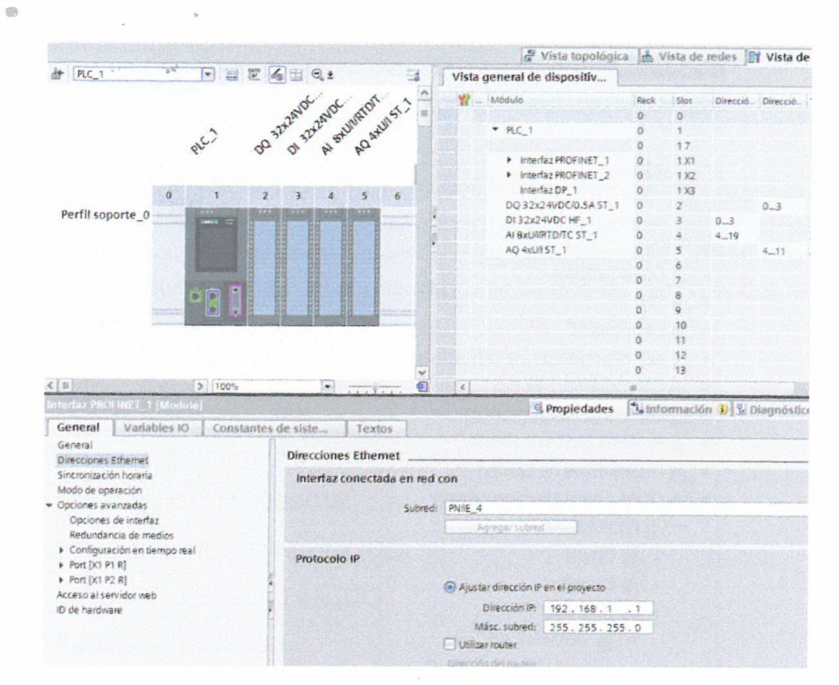


Figura 135

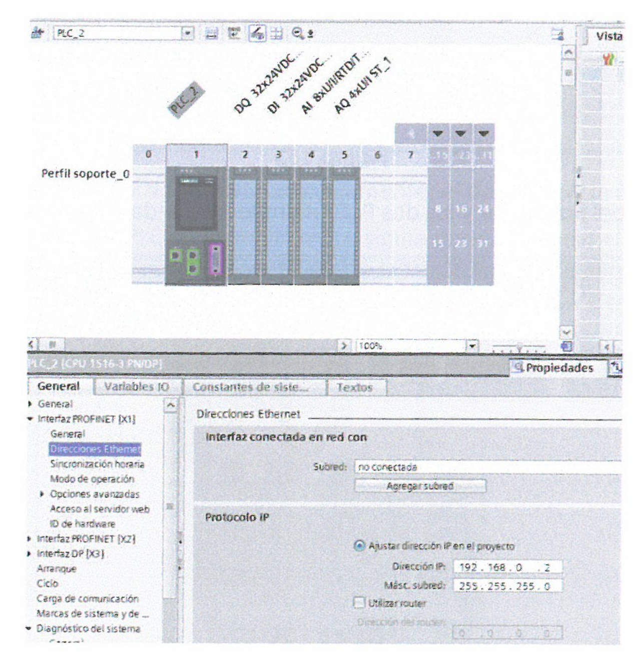


Figura 136

El siguiente paso consiste en ofrecer al acoplador las direcciones por donde va a enviar y por donde va a recibir al otro grupo de la red. Para ello deben tomarse los módulos que se deseen del catálogo y colocarlos sobre la parte X1 y X2 del acoplador. Resulta muy importante cruzar esas direcciones entre la parte X1 y X2 del acoplador. En las figuras 137 y 138 se representan las dos partes del acoplador. Hay que tener en cuenta que todavía no se ha realizado la conexión entre los PLC y la parte X1 y X2 del acoplador. Por eso, al colgar los módulos de E/S, no aparecen las direcciones que les corresponde; tampoco deja insertarlas de forma manual.

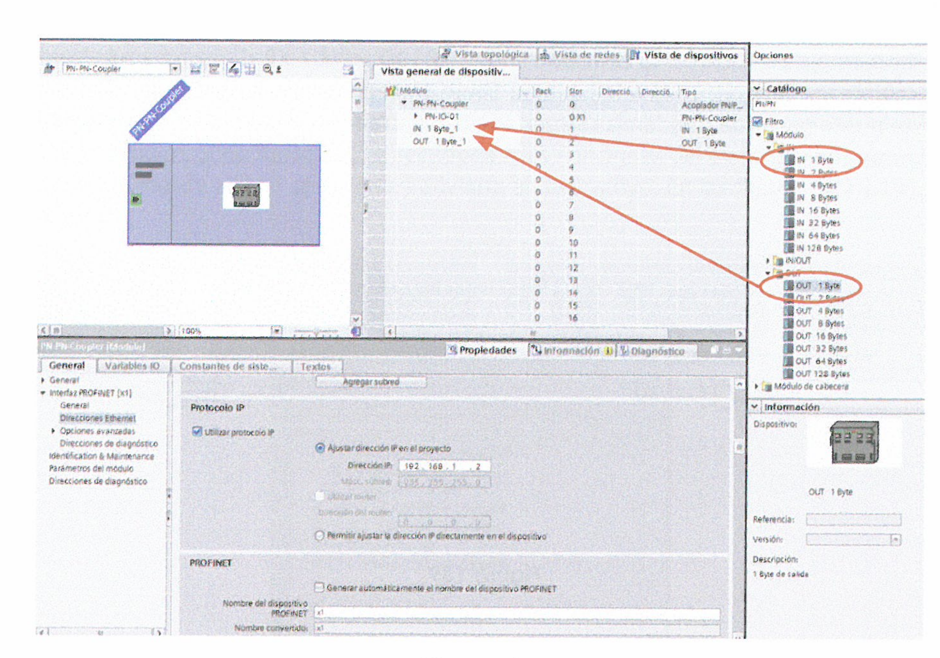


Figura 137

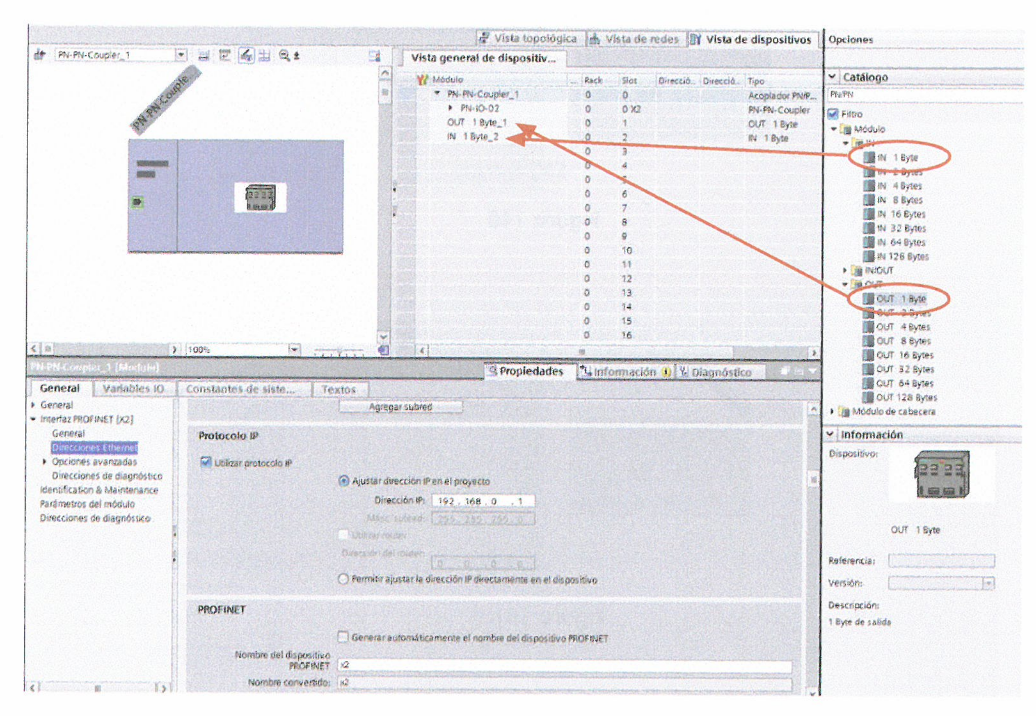


Figura 138

Ha llegado el momento de establecer la conexión entre PLC1-X1 y PLC2-X2. Si observamos los dos lados del acoplador, ya saldrán las direcciones que le ha dado TIA PORTAL. Esto puede verse en la figura 139. Se debe revisar bien este punto para que queden las direcciones de entrada y salida cruzadas, como puede comprobarse en las figuras 140 y 141.

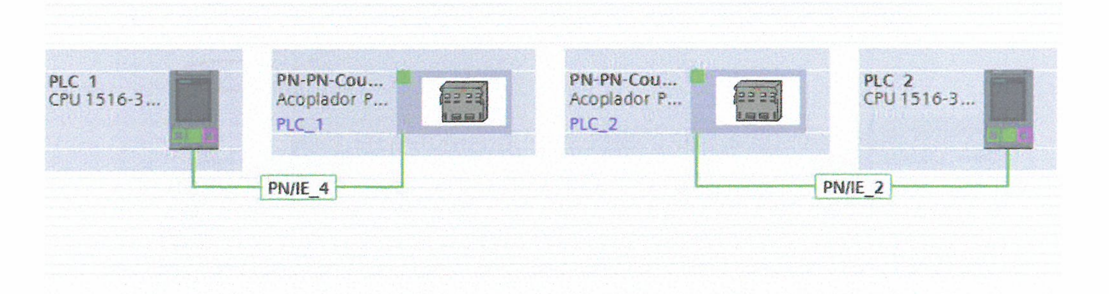


Figura 139

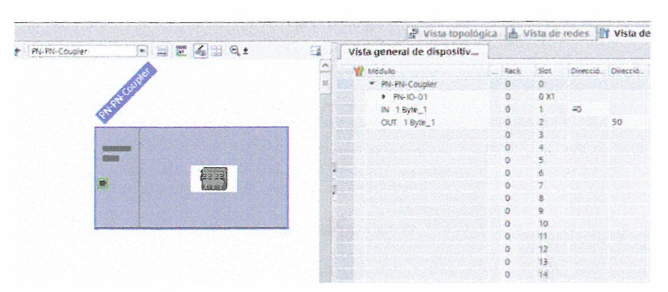


Figura 140

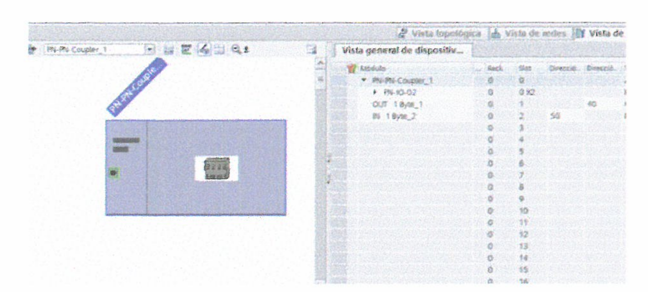


Figura 141

Por último, se realizará una última comprobación, de importancia vital. Se trata de los nombres de Profinet que presentan los dos lados del acoplador. Cada lado debe poseer distintos nombres y, lo que es más importante, han de ser los mismos en la configuración que se le ha dado en TIA PORTAL y la que se halla realmente grabada en su memoria física.

Para ello, se puede acceder a la red y comprobar si el nombre con el que cuenta en el proyecto de TIA PORTAL es el mismo que aparece al acceder de forma *online*. Como ya se ha dicho en diversas ocasiones, en la ventana del árbol del proyecto, accedemos *online* con la interfaz de la tarjeta de red. Se ha de comprobar que los nombres coinciden. Si no fuese así, se accedería a las *Funciones* del dispositivo a través de la opción *Online y diagnóstico* y, desde allí, se cambiaría y enviaría al dispositivo. En las figuras 142 y 143 se recuerda este procedimiento.

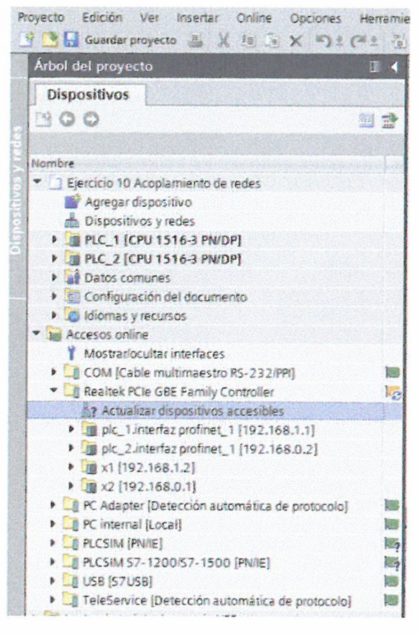


Figura 142

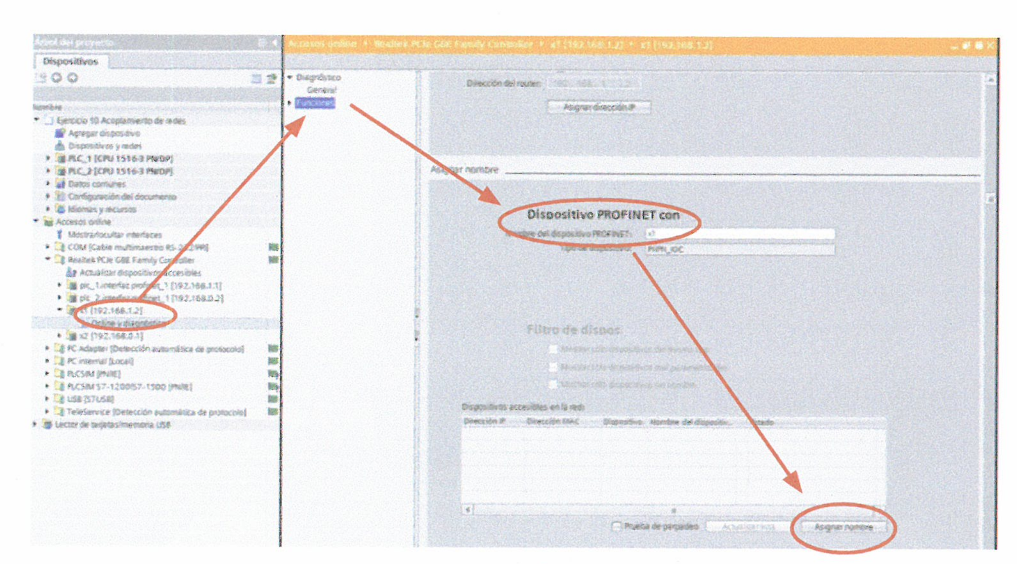


Figura 143

Con esta última comprobación, la parte correspondiente a la configuración ya se encuentra concluida. Ahora se necesita comprobar que existe intercambio de información entre los dos grupos y, para ello, se habrá de cargar en cada PLC las siguientes instrucciones:

OB1 PLC1:

```
L    "Entrada_B0_PLC1"           %EB0
T    "Salida de X1 hacia X2"     %AB50

L    "Entrada de X1"             %EB40
T    "Salida_B0_PLC1"            %AB0

```

OB1 PLC2:

```
L    "Entrada_EB0_PLC2"          %EB0
T    "Salida de X2 hacia X1"     %AB40

L    "Entrada X2"                %EB50
T    "Salida_AB8_PLC2"           %AB8

```


Unidad 5 Ejercicios prácticos en SCADAS e Internet



En esta unidad veremos:

13. Ejercicio de SCADA en Profinet
14. Web de usuario con PLC S7-1516
15. Web de usuario con PLC S7-1516 utilizando comandos AWP
16. Página web de una CPU 1500

13. Ejercicio de SCADA en Profinet

Enunciado

En este ejercicio se va a realizar la configuración de un SCADA. Se utilizará el SCADA DSC de National Instruments que corre bajo la plataforma de programación LabVIEW. Se establecerá la comunicación del SCADA/PC con el PLC S7-1500 utilizando OPC-Server y OPC-Cliente y Ethernet.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. *Software* SIMATIC NET de Siemens.
3. *Software* SCADA, **Datalogging and Supervisory Control (DSC).**
4. Tarjeta de red Ethernet en el PC.
5. OPC servidor/cliente.
6. Un cable de Profinet para las conexiones.

Realización

Lo interesante de este ejercicio es el uso de una plataforma SCADA de un fabricante diferente al del PLC. El PLC es de Siemens, igual que su plataforma de configuración y programación, y National Instruments en el caso del SCADA. Para ello se va a utilizar el *driver* OPC y también será necesario emplear algunas herramientas imprescindibles que se encuentran dentro del *software* de Siemens denominado SIMATIC NET.

El esquema de realización del ejercicio va a ser el siguiente:

1. Configuración del PLC y **su envío al PLC**.
2. Inclusión de la *Estación PC* (ordenador) en el proyecto.
3. Addición de OPC y tarjeta de red a la *Estación PC.*
4. Creación de la conexión del tipo S7 entre el PLC y la *Estación PC*.
5. Configuración del ordenador mediante la aplicación *Station Configuration Editor* del *software* SIMATIC NET.
6. Envío de la configuración creada a la tarjeta de red.
7. Uso del cliente OPC SCOUT V10.
8. Configuración del SCADA DSC.
9. Desarrollo de un programa para comprobar el intercambio de datos.

Respecto al *software* de SIMATIC Net, se ha utilizado la versión V12+SP2 y de OPC, la versión 8.2.

Con el ejercicio se pretende introducir en el sistema de proceso el propio PC para monitorizar, controlar y registrar datos. Si se desea comunicar el PLC con el PC como un participante más en el proceso de automatización, será necesario un *driver* que favorezca la conexión entre los dos. Y ya no solo se trata del PC; también se ha de conseguir que la conexión se establezca con el SCADA que corre en dicho PC. Además, ese *driver* debe servir para poder conectar cualquier dispositivo de cualquier marca comercial. Esa función la desempeña el OPC. Este es un potente *driver* al que se le puede denominar *driver universal*, porque vale para cualquier

dispositivo que disponga y trabaje con OPC. Hoy día se está en un proceso de evolución de OPC y se utiliza el denominado OPC UA, un OPC clásico ampliado a más tipos de datos. Para entender el alcance del uso de OPC, solo basta con recordar qué sucedía, hasta no hace mucho tiempo, cuando se compraba una impresora para conectar en nuestro PC. Se necesitaba el *driver* de la impresora adecuado al sistema operativo con el que funcionaba en el PC y eso obligaba a los fabricantes a contar con los *drivers* para todos los sistemas operativos que existían en el momento. Si hubiera existido OPC, no existiría un *driver* para cada sistema operativo. Si el fabricante y el creador del sistema operativo dispusieran de OPC, no habría que preocuparse por nada más.

OPC es OLE ***(Object Linking and Embedding)*** para control de procesos. Pero ¿qué es OLE? Tanto OLE como OPC son aplicaciones creadas por Microsoft. Explicar estas cosas no resulta siempre fácil si no se está muy formado en temas de comunicaciones informáticas. Ejemplificaremos más adelante dónde se puede ver lo que significó el uso de OLE. Hoy todo el mundo cree que siempre existió la funcionalidad de copiar de un documento cualquiera a otro documento: el *copia y pega*. Esta acción que todos realizamos de forma automática no existía en la década de los ochenta; apareció en la de los noventa. La comodidad de poder copiar un texto y pegarlo en otro conforma alguna de las cosas que consiguió la aplicación OLE. Lo mismo sucede con el *arrastrar y soltar*. Ese concepto de universalidad y compatibilidad es el que se quería adaptar en la industria y, por ello, nació OPC.

Se va a comenzar con la configuración del PLC. En este caso, se necesita que el PC posea una dirección dentro del mismo grupo que la tarjeta de red del PC; por ejemplo, si la dirección de la tarjeta de red del PC es la 10.0.127.100 y su máscara de subred, 255.255.255.0 el PLC deberá presentar otra dirección, pero dentro de ese grupo y, por lo tanto, la misma máscara de subred. Se le va a poner al PLC la dirección 10.0.127.200.

Para saber la dirección de la tarjeta de red del ordenador, se ha de ejecutar el comando «cmd» en inicio. Esto dependerá del sistema operativo que se disponga. Una vez ejecutado el comando «cmd» se abrirá la consola *MsDos* y allí se teclea el comando «IPconfig», de forma que aparezca la dirección IP del ordenador ***(IPv4)***, como se ve en las figuras 144 y 145.

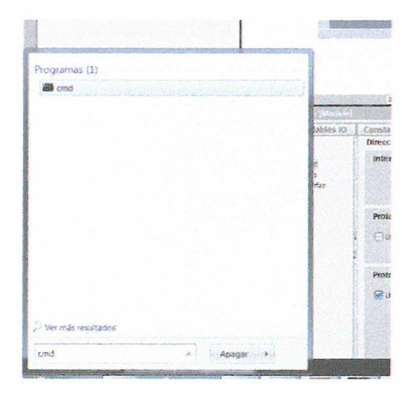


Figura 144

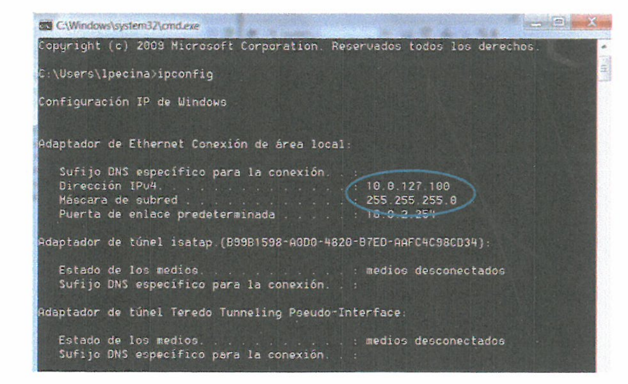


Figura 145

Otra cosa que se debe hacer con el PLC es habilitar la opción que permite usar OPC. Es la misma que ya se había seleccionado para usar los bloques GET y PUT. En la figura 146 se recuerda esta situación.

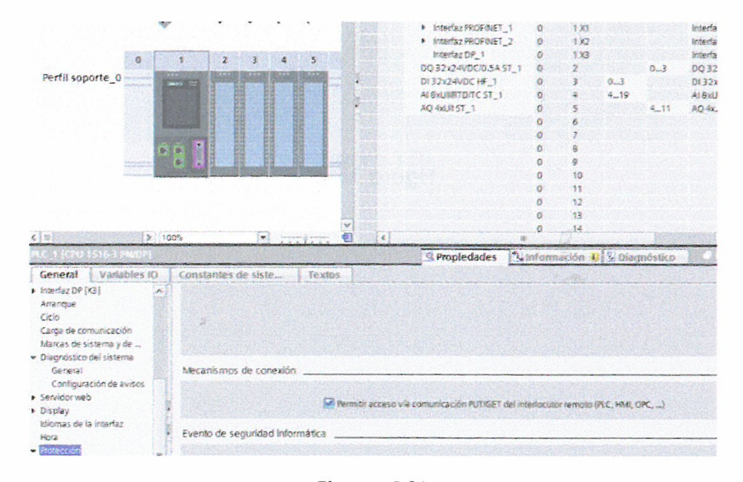


Figura 146

Tras cambiar la dirección del PLC, se envía la configuración al autómata. Con ello se concluye el punto 1 del guion de configuración.

Posteriormente, se carga el PC en *Vista de redes*. Se denomina *Estación PC* y, dentro de ese ordenador, se encuentran dos elementos, uno de *hardware* y otro de *software*. El *hardware* corresponde así de la tarjeta de red que lleva el ordenador y el *software* pertenece al *OPC Server*. Ambos se sitúan en el catálogo. Como tarjeta de red del ordenador se pone una interfaz general de Ethernet, denominada *IE general*. Existen varias versiones tanto de *OPC* como de *IE general*. Utilizamos la versión 8.2 para el *OPC* y la versión 6.2 para la *IE general*.

En la figura 147 se muestra dónde localizar la *Estación PC*.

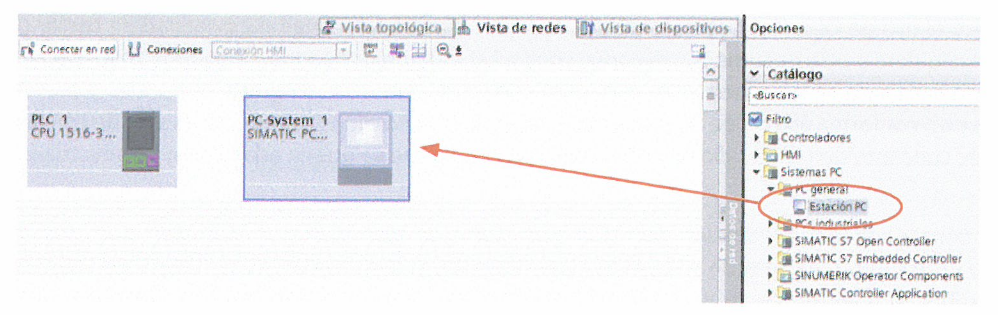


Figura 147

En las figuras 148 y 149 se indica dónde se encuentran el *OPC Server* y la *IE general* (desde *Vista de dispositivos*). Observe en el área inferior derecha la versión y seleccione la adecuada.

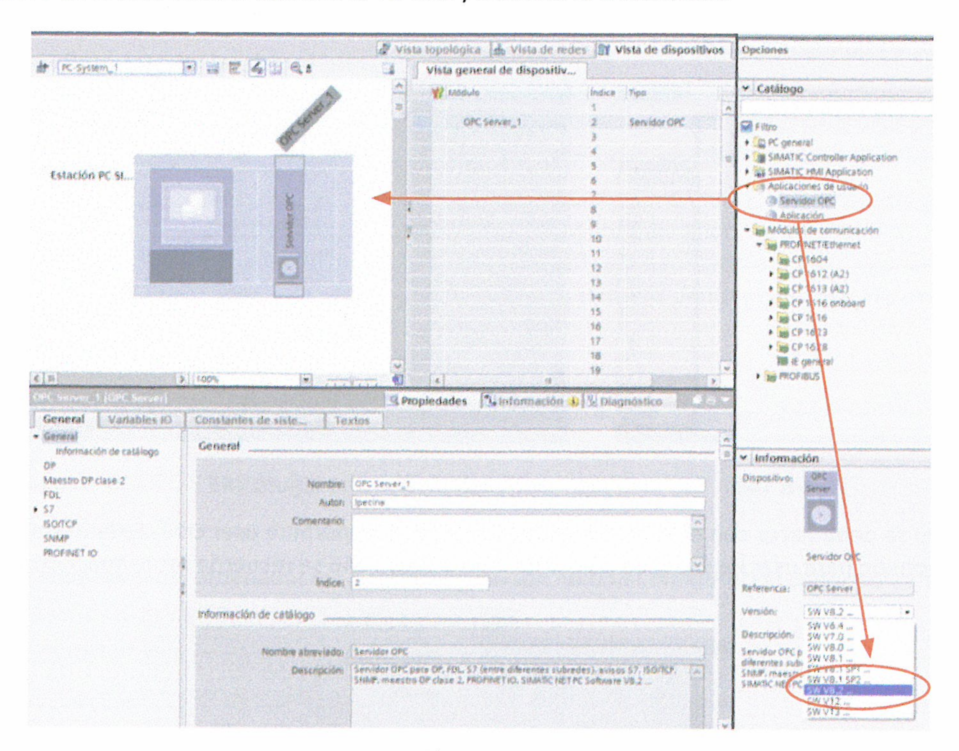


Figura 148

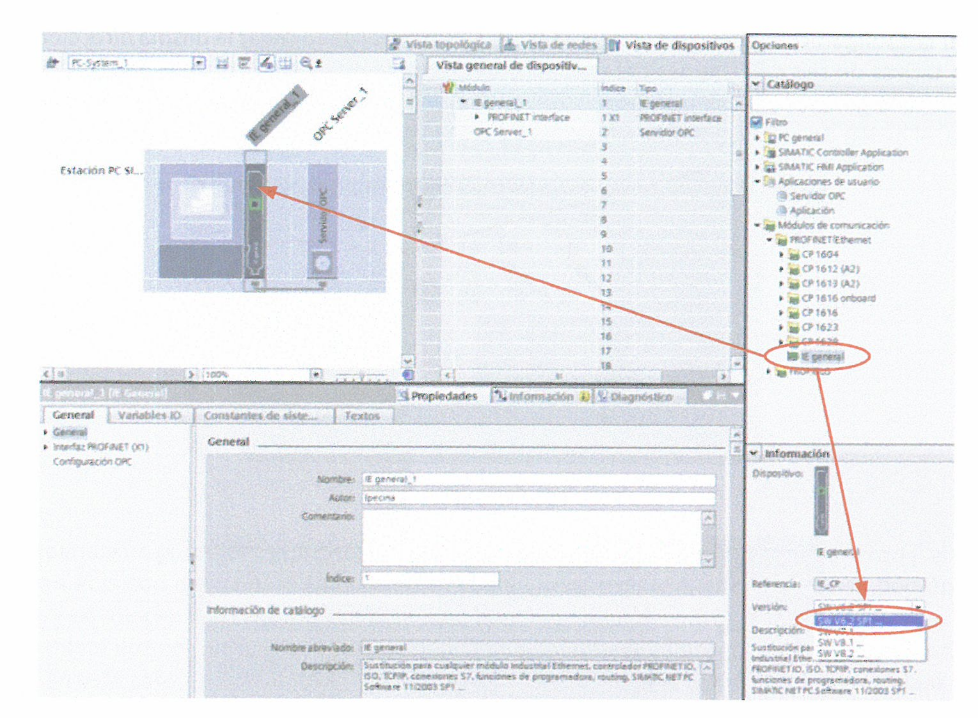


Figura 149

En la figura 150 puede verse cómo quedará la *Estación PC*.

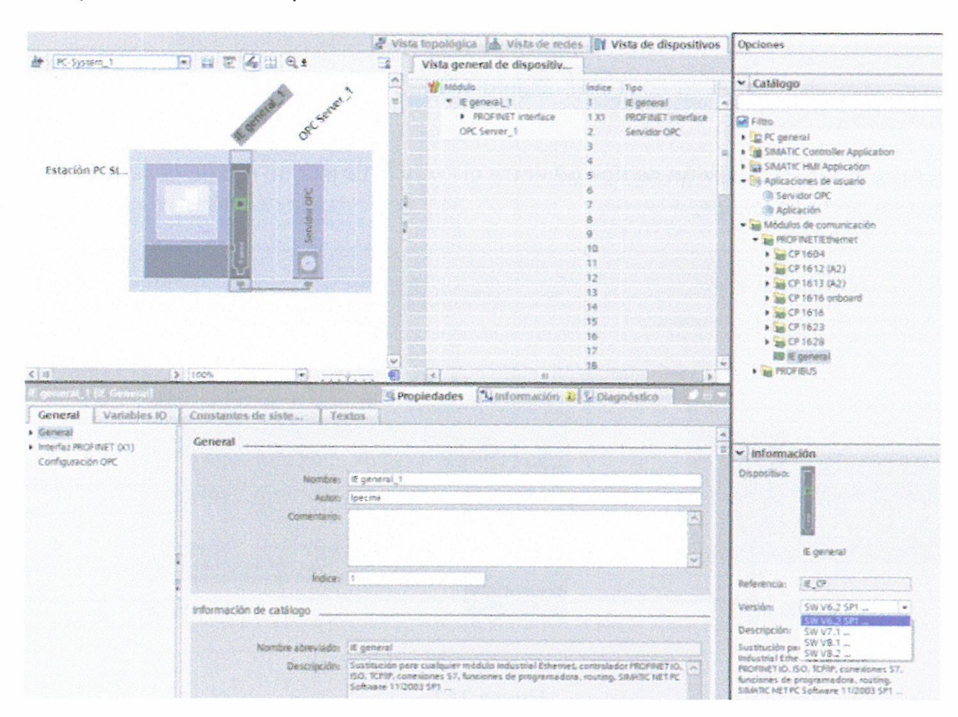


Figura 150

Para terminar la configuración de la *Estación PC*, la tarjeta *IE general* debe llevar la misma dirección que tiene realmente la tarjeta de red de Ethernet del ordenador. En el caso de este ejercicio, es la 10.0.127.100 y se cambia desde *Vista general de dispositivos* en *Propiedades*, como se ve en la figura 151. Obsérvese en esa misma figura, el **orden de colocación** de la IE general y el OPC porque será necesario para el siguiente paso.

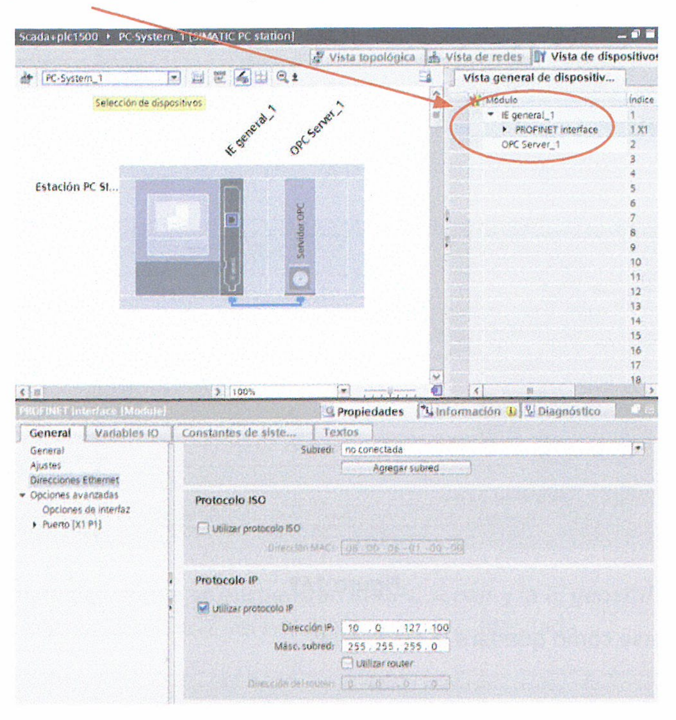


Figura 151

A continuación, se realiza la conexión de tipo S7 entre el PLC y la tarjeta de red *(IE general)*. En la figura 152 se representa esta parte concluida.

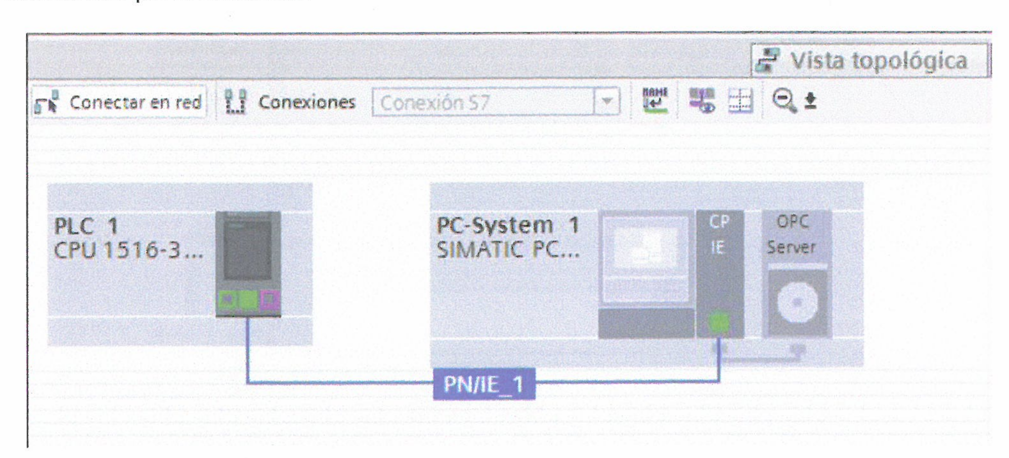


Figura 152

Con estos pasos se llega al punto 4 del guion. En este momento, se ha de emplear una aplicación que contiene el grupo de programas denominado SIMATIC NET. No será posible llevarlo a cabo si no se dispone de este *software*. La aplicación se denomina *Station Configuration Editor.* Al instalar el *software* SIMATIC NET, saldrá su icono en la esquina inferior derecha del ordenador (figura 153).

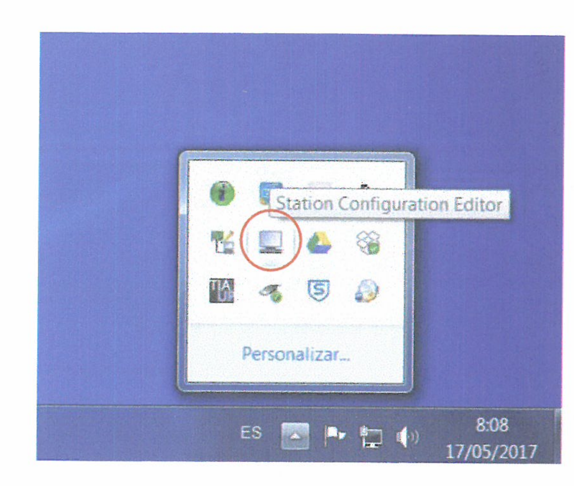


Figura 153

Al abrirlo, aparece un grupo de celdas sobre las que se colocará la tarjeta de red *(IE general)* y el OPC Server. Pulsando con el botón derecho en cada una de las filas, surgirán los elementos que se podrán colocar. Se seleccionan y se colocan en el **mismo orden** en que han sido situados en TIA PORTAL (véanse las figuras 154, 155 y 156). Habrá que comprobar que la dirección IP que aparece al colocar la tarjeta sea la misma que tiene realmente; en este caso, la IP es la 10.0.127.100. Si no lo fuera, se deja como está y, al terminar de cargar la tarjeta y el OPC Server, se guarda el proyecto y se reinicia el ordenador; después se comprueba (con doble pulsación sobre la tarjeta) si constituye la dirección correcta. Otro aspecto que se ha de tener en cuenta es que el nombre que aparece en *Station Configuration Editor* sea el mismo que tiene adscrito en el proyecto de TIA PORTAL. Si no fuera así, habría que ponerlo igual.

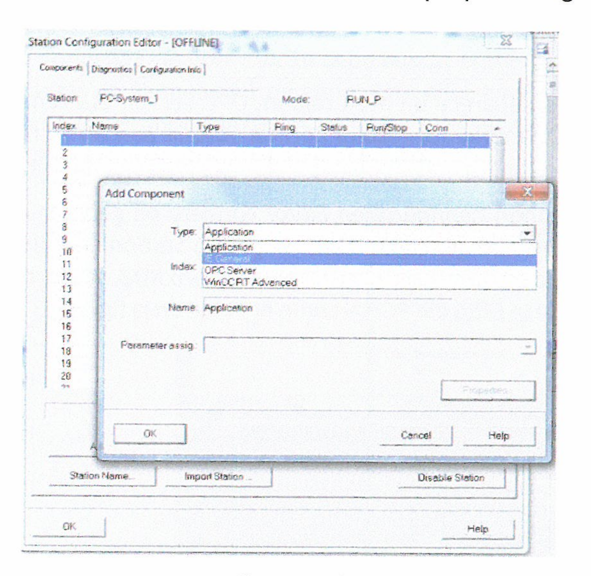


Figura 154

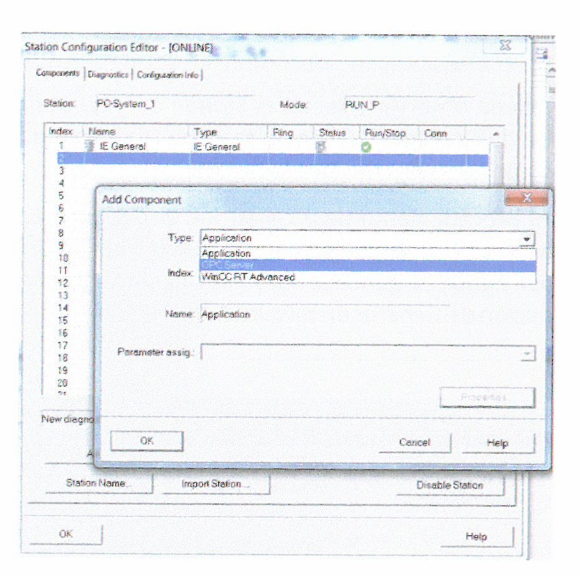


Figura 155

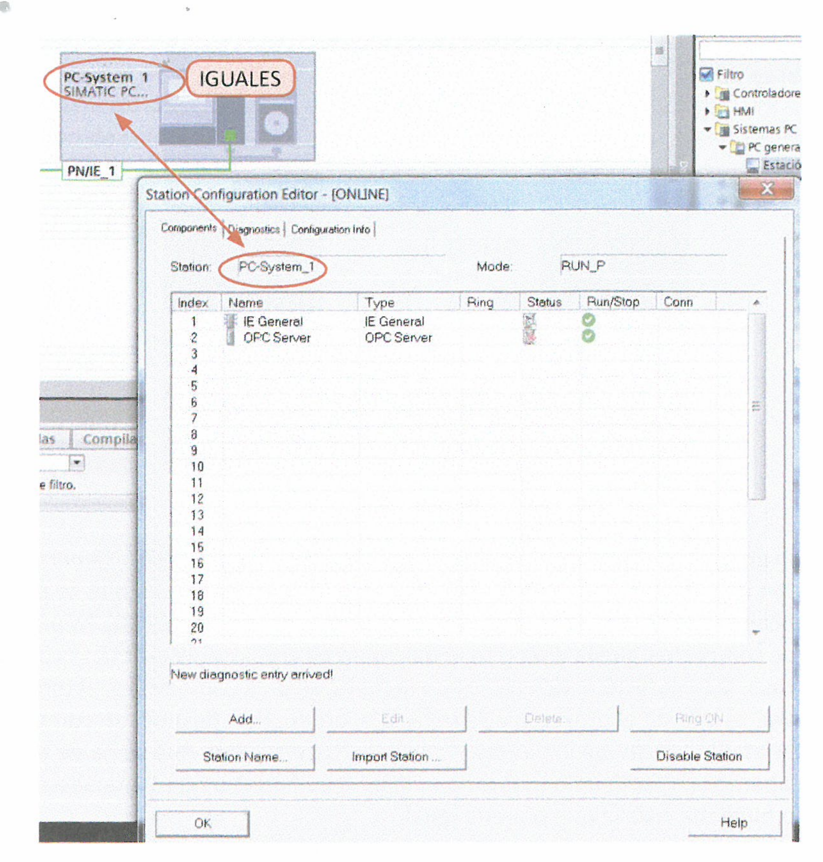


Figura 156

Se puede observar que, en el *Station Configuration Editor,* aparece un aspa roja en OPC Server. Esa señal debe desaparecer cuando se termine la configuración. De momento resulta normal porque aún no se ha enviado la configuración a la tarjeta.

Otro procedimiento para cargar los dispositivos en el *Station Configuration Editor* consiste en generar el fichero XDB. Si se procede así, se puede importar el fichero XDB generado desde el propio *Station Configuration Editor,* cargándose automática y correctamente cada elemento. Para generar el fichero XDB, se marca la opción de *Generar archivo XDB* desde las propiedades de la *Estación PC*, tal como aparece en la figura 157.

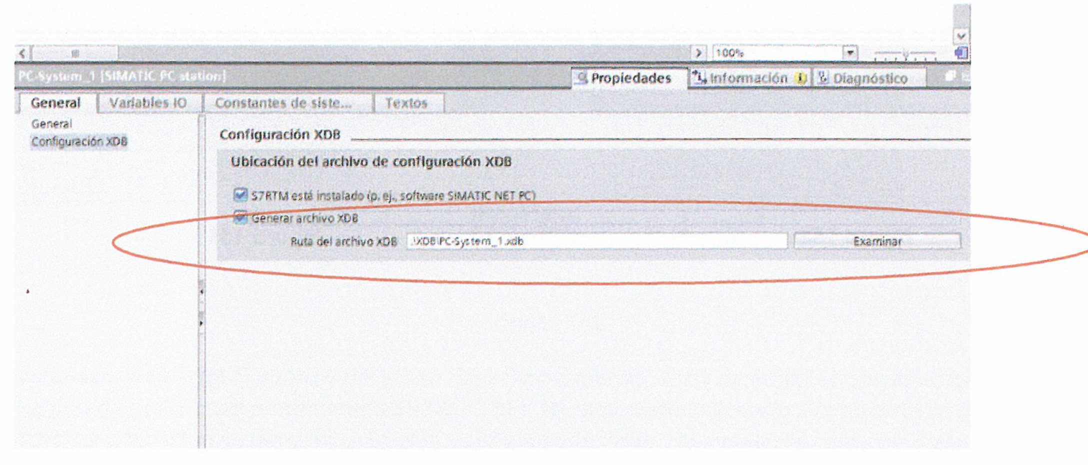


Figura 157

Ha llegado es el momento de enviar la configuración a la tarjeta. Se hace igual que si fuera el PLC: cuando salga la pantalla principal de envío, se selecciona *Buscar.* Puede que no aparezca. Si ocurre, se teclea dos veces en el campo la dirección que se está buscando, en este caso la 10.0.127.100, y se da a *Buscar* nuevamente para que la encuentre. Una vez hallada, se carga. Véase la figura 158. Al finalizar, ya se estará en condiciones de pasar al punto 7.

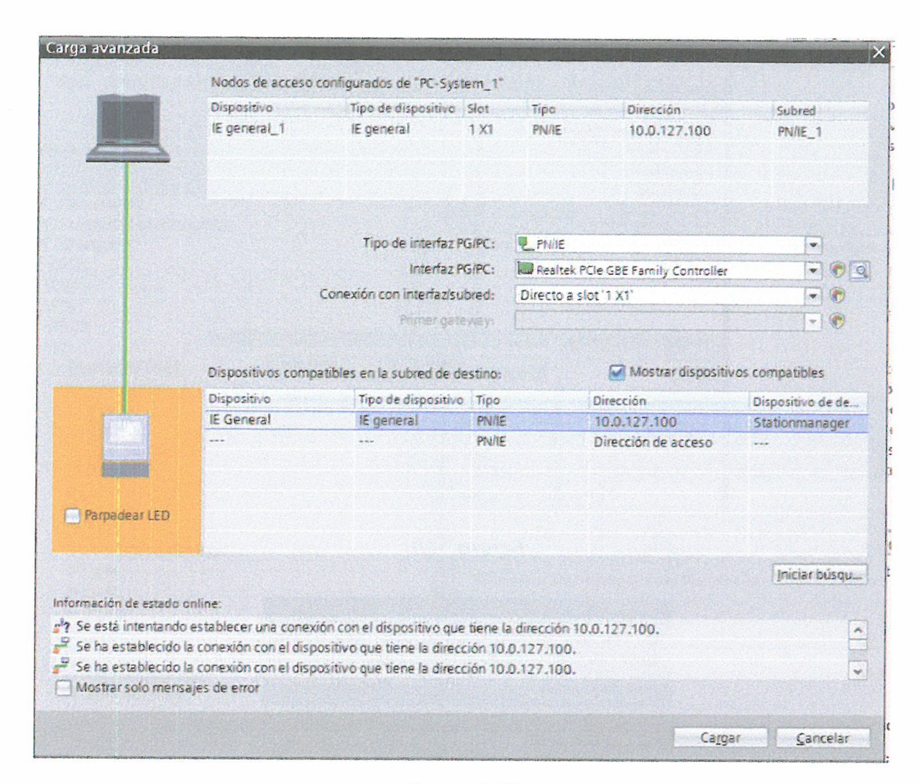


Figura 158

Para el paso siguiente se necesita otro programa que incluye SIMATIC NET. Se trata del SCOUT; en este caso se emplea la versión V10. Se debe buscar entre los programas de *Inicio del ordenador*, en la carpeta de programas de Siemens Automation. Este programa es un OPC cliente, como si fuera el SCADA y, de hecho, se van a poder monitorizar las variables que se seleccionen. También aquí se seleccionarán las variables que posteriormente se utilicen en el SCADA.

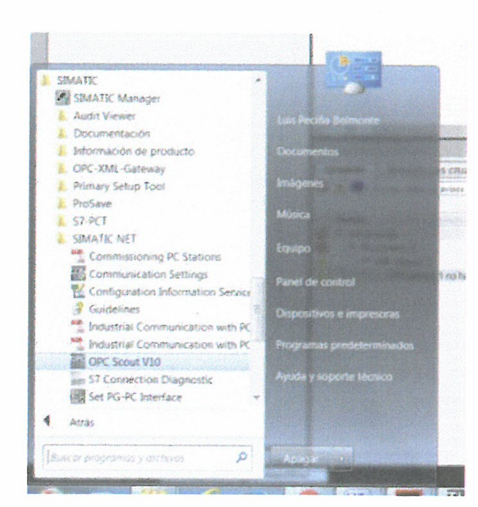


Figura 159

Una vez abierto, el SCOUT adquiere la apariencia que se refleja en la figura 160. Tiene que aparecer el servidor que hemos configurado: *OPC.SimaticNet*. Se abre y se conecta con él pulsando doblemente. Aparecerán en *Objetos* (dentro de la conexión *S7_Connection_1*) las variables del PLC (aquí llamadas en lenguaje internacional I, Q... en vez de E, A...) (figura 161).

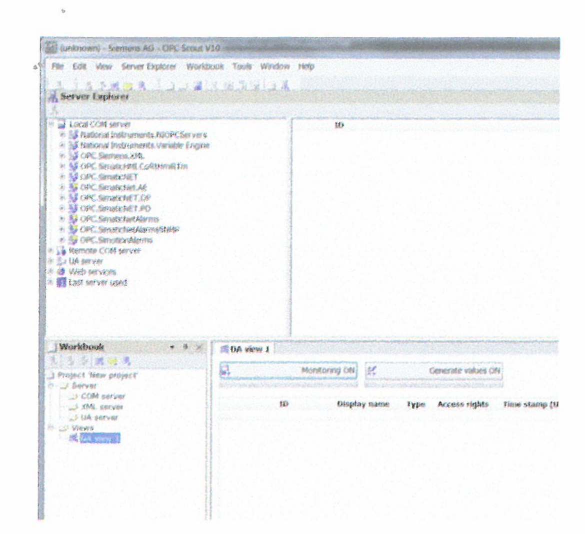


Figura 160

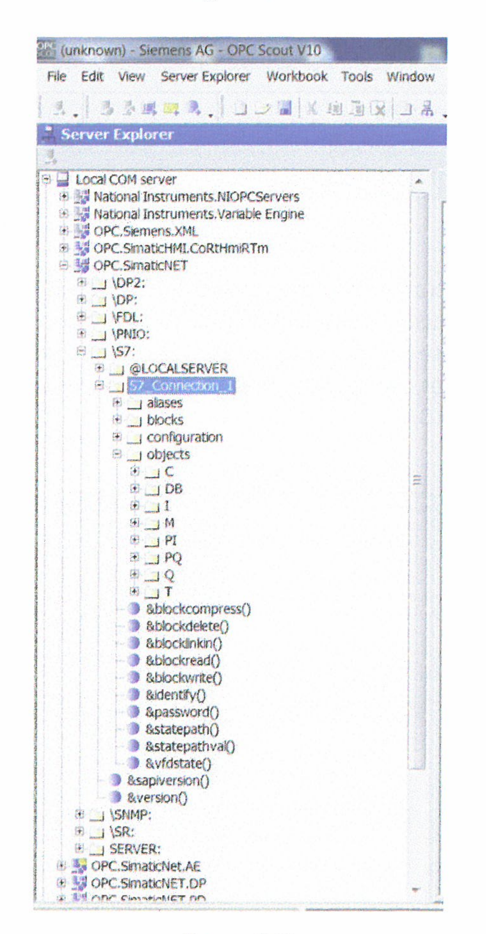


Figura 161

Ahora se seleccionan las variables que se desean utilizar y que luego se emplearán en el SCADA. Aquí se selecciona una entrada digital y una salida digital, ambas de tipo bit. El tipo de dato para el bit es la **X**. Aparece un cuadro para escoger el tipo de dato de la variable y su dirección. Se le puede poner también un alias. En *Number of value,* se escribe un 1 si solo se desea un bit. Si se quieren 8 bits, se colocaría 8 y, de esta forma, luego se puede utilizar como *array* (matriz). Véase la figura 162.

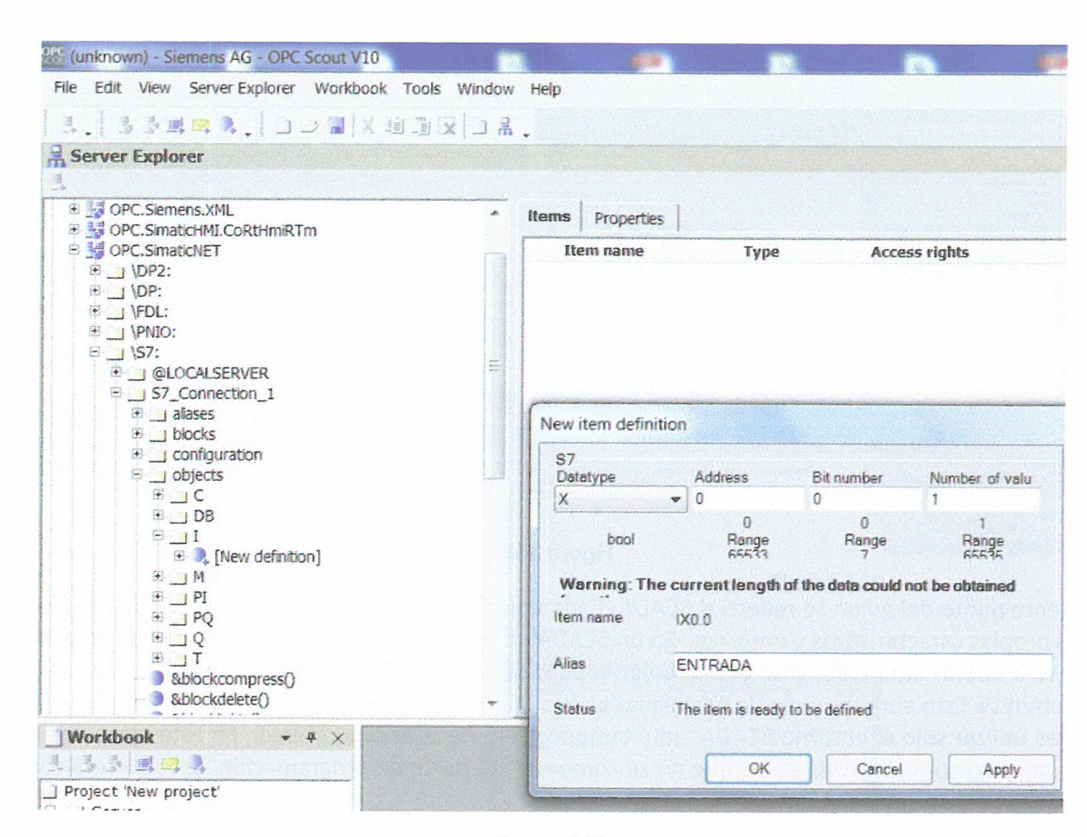


Figura 162

Se hace lo mismo para una salida. Aparecerán las variables seleccionadas en el árbol de variables. Ahora se debe crear un nuevo visor de variables DA y ya se podrá monitorizar llevando las variables a la zona del visualizador. Se observa en las figuras 163 y 164.

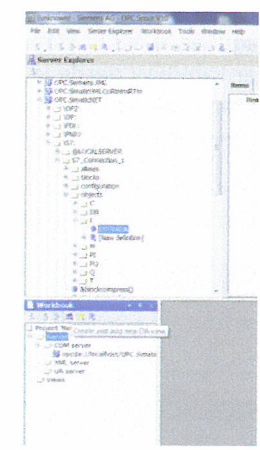


Figura 163

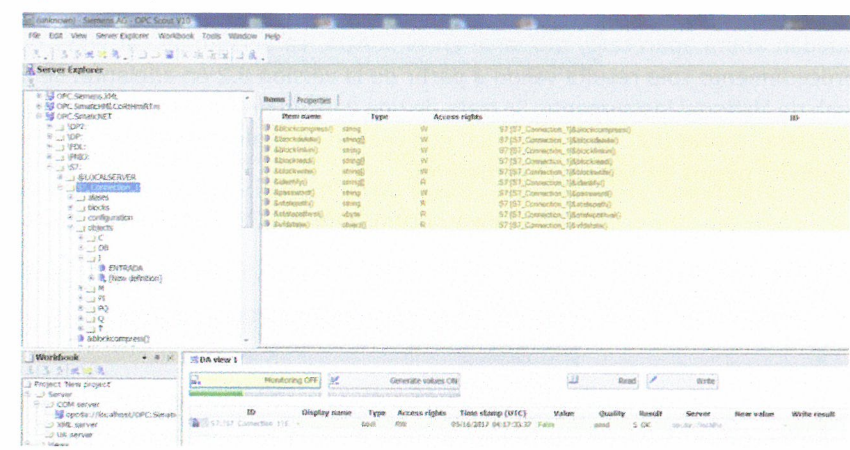


Figura 164

Para monitorizar las entradas, se activa *Monitoring ON* y, al pulsar sobre la entrada en el PLC, se verá reflejado en el SCOUT. Si se desea monitorizar una salida, se pulsa en *Nuevo valor* y se selecciona el deseado. A continuación, se activa *Escribe* y la salida del PLC cambiará. Resulta muy importante que la calidad de la conexión se encuentre en *Good.* Si no fuese así, se guarda y se cierra todo con el fin de reiniciar el ordenador. Seguidamente, se vuelve a comprobar si la conexión ya es *Good* (figura 165).

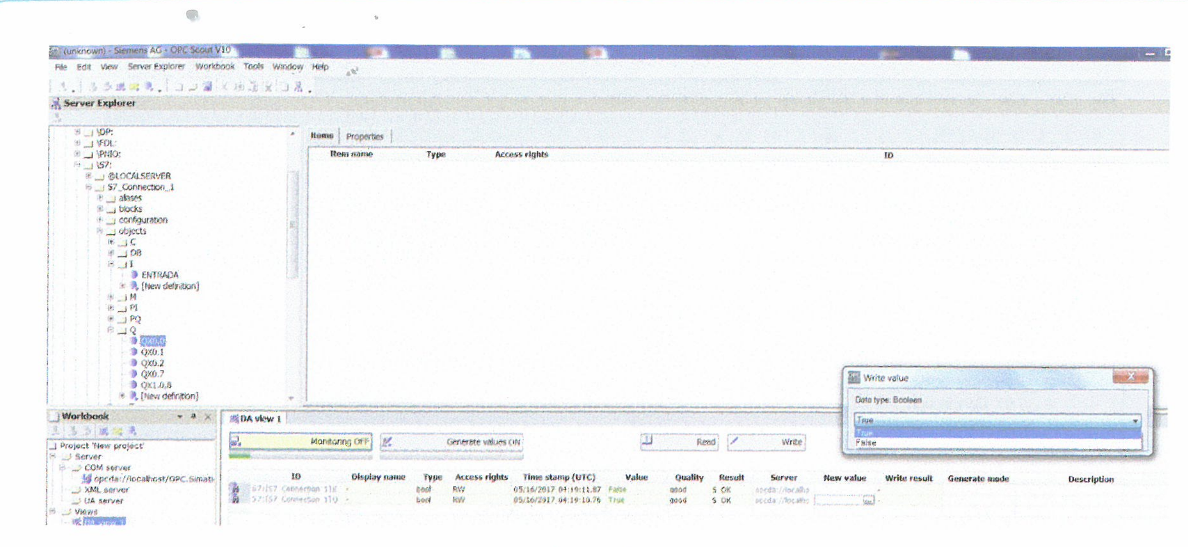


Figura 165

El siguiente punto del guion se refiere al SCADA. Cada uno de los SCADA disponibles en el mercado cuenta con sus propias características y entornos. En un SCADA no hay que programar, solo configurar. En el SCADA que se va a utilizar aquí, DSC, sí se puede programar, si se desea, porque funciona en un entorno denominado LabVIEW. Esto abre un sinfín de posibilidades que lo hace extremadamente potente. Pero, si lo que se quiere es utilizar solo el entorno SCADA, aquí tampoco será necesario programar. En este ejercicio se va a utilizar tan solo como SCADA, por lo que no se comentará la parte de programación. Sí se deberá conocer mínimamente el entorno del SCADA.

El SCADA debe poder trabajar bajo OPC; en este caso, será un cliente OPC. Hoy día todos tendrían que mostrar esa capacidad. El procedimiento que se va a describir en este ejercicio y para este SCADA resulta similar en todos los SCADAS comerciales. Lógicamente habrá que estudiar en otro SCADA cómo y dónde se pueden llevar a cabo los pasos que se explican a continuación.

En primer lugar, se prepara el terreno para que las variables que se han seleccionado mediante el SCOUT puedan ahora aparecer y utilizarse en el SCADA. La versión de LabVIEW empleada aquí es la V8.5. Este procedimiento solo resulta válido a partir de la versión 8.2 en adelante. Se puede bajar LabVIEW desde la web de National Instruments. Se trata de una versión de evaluación. Una vez instalado LabVIEW, se deberá instalar el *toolkit* DSC. Para descargar una versión de evaluación de LabVIEW y DSC, se puede hacer desde la siguiente página de National Instruments:

http://www.ni.com/gate/gb/GB_EVALTLKTINDCTRL/ESA

Una vez abierto LabVIEW, se selecciona un proyecto vacío *(Empty Project)* (figura 166).

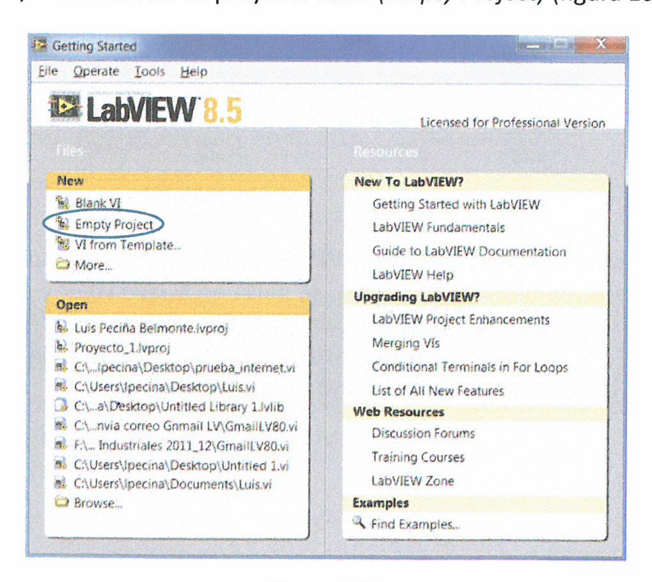


Figura 166

A continuación, se abre un nuevo servidor en *My Computer*. El servidor será el OPCSimaticNET_1. Para ello deben seguirse los pasos de las figuras 167-169 (alguna tarda un tiempo en salir):

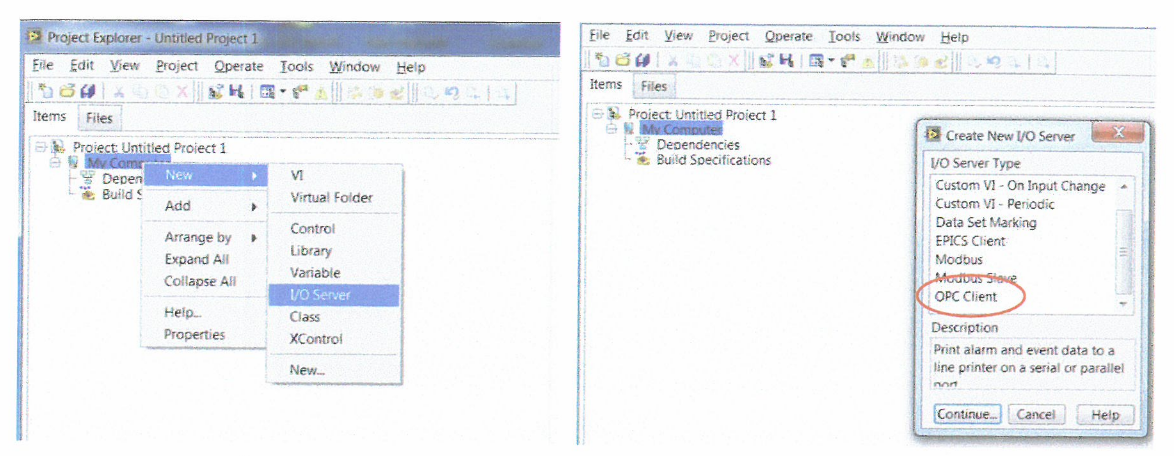


Figura 167 Figura 168

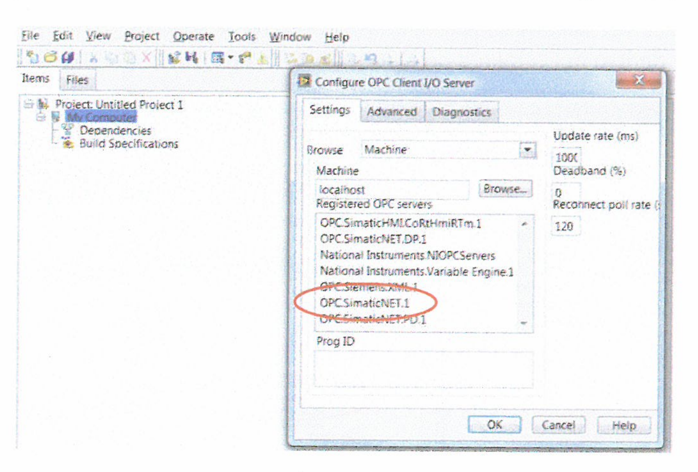


Figura 169

Después, en la librería creada del servidor, se añaden todas las variables que se desean utilizar en el SCADA y que se servirán vía OPC desde el PC. Los pasos se indican en las figuras 170 y 171. Hay que esperar hasta que salgan las variables.

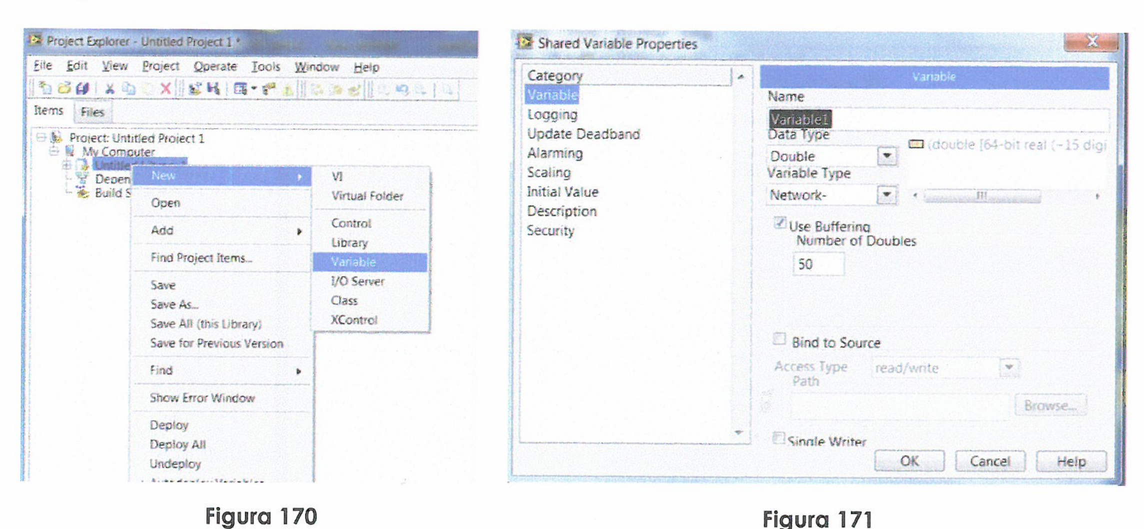


Figura 170 Figura 171

En la ventana de la figura 171, se debe dar un nombre a la variable y activar *Bind to Source* («enlazar a la fuente») para vincular esta variable a la variable fuente que proviene del OPC Server. Seguidamente, se navega *(Browse...)* para encontrar la variable que procede del servidor OPC (figura 172).

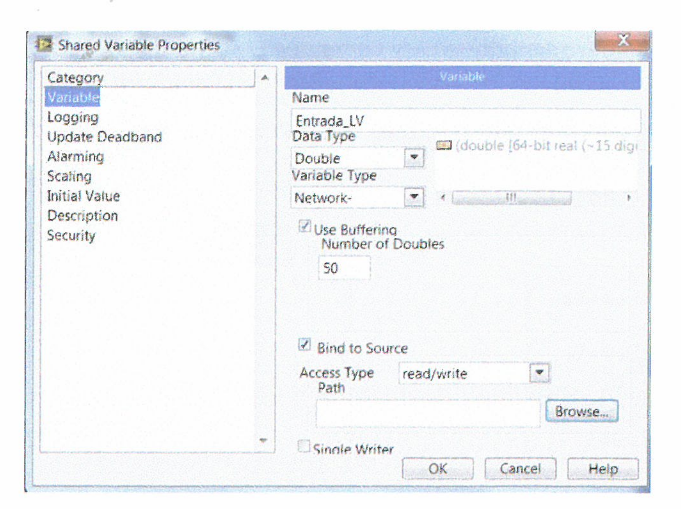


Figura 172

Se navega en el menú hasta encontrar la conexión S7_Connection_1, así como la variable deseada, bien en *Alias,* bien en *Objetos*. Una vez hallada se selecciona y se acepta la ventana donde ya aparecerá dicha variable de modo que queda dentro del catálogo de las variables del proyecto (figuras 173-175).

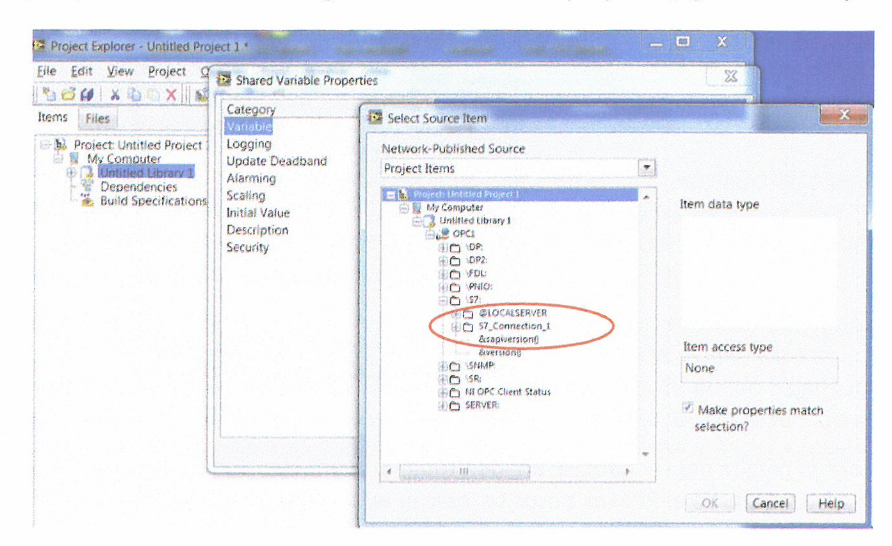


Figura 173

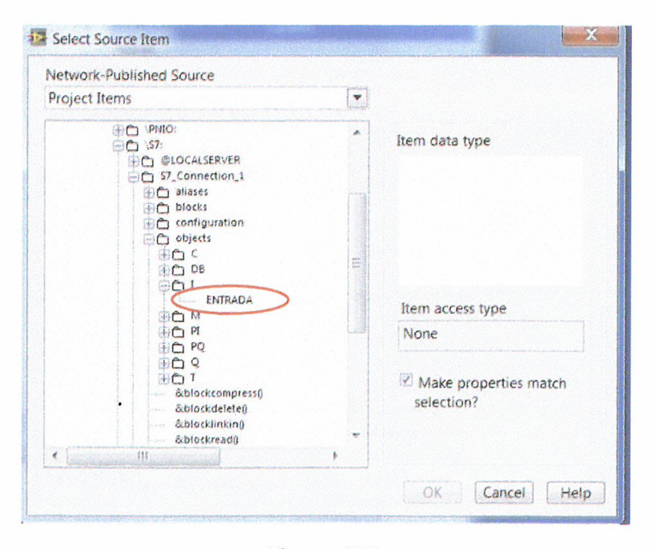


Figura 174

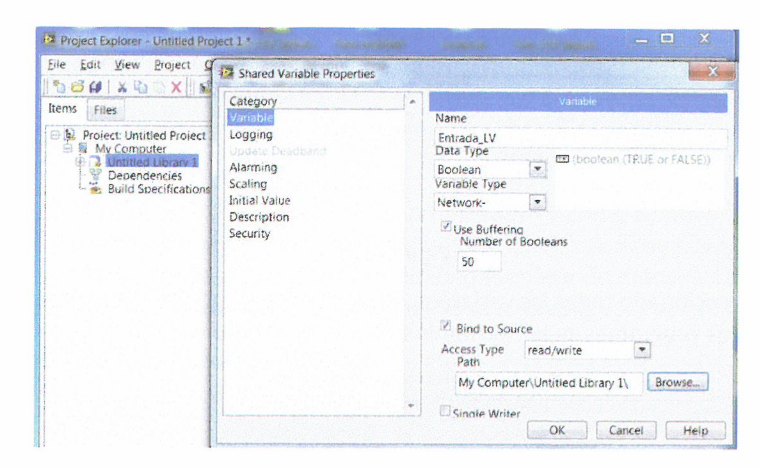


Figura 175

Tras seleccionar todas las variables, deberán aparecer como se muestra en la figura 176.

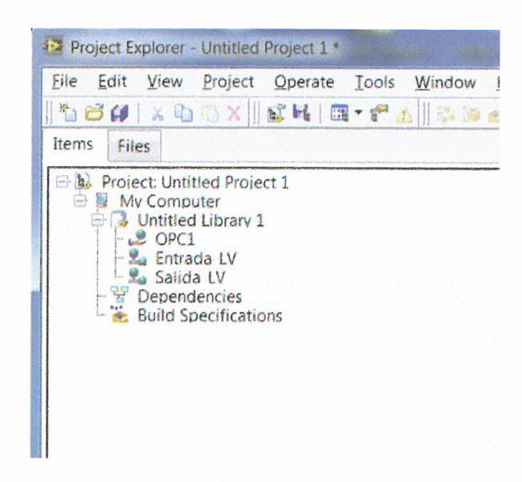


Figura 176

Si existen muchas variables, como es habitual, resultaría muy tedioso añadirlas una a una. Para ello se puede escoger la opción *Create Bound Variables* («crear variables enlazadas»). Esto se hace en vez de recurrir al paso de selección individual de variables. De esta forma, como se indica en la figura 177, se seleccionan todas las variables de una vez al añadirlas a la parte de la derecha de la figura 178. El punto de partida de esta opción se aprecia en la figura 176. Posteriormente se pueden editar las variables, por si se desea cambiar el nombre, por ejemplo. En la figura 179 se expone la ventana de edición de las variables.

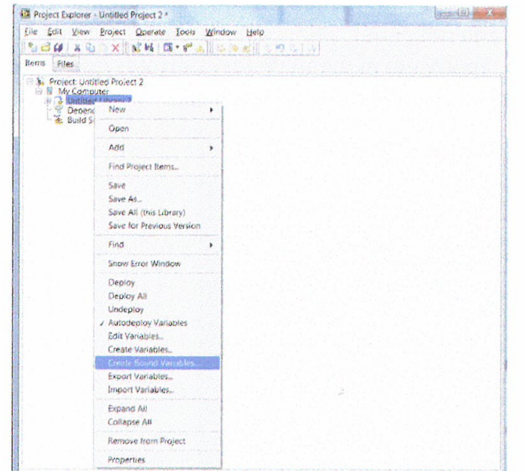


Figura 177

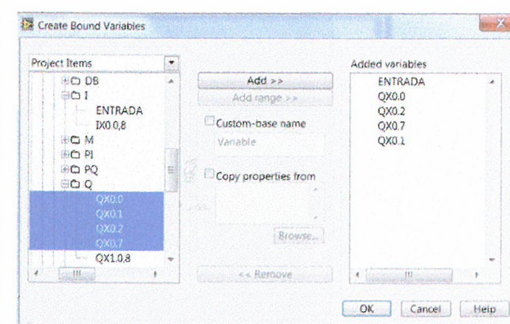


Figura 178

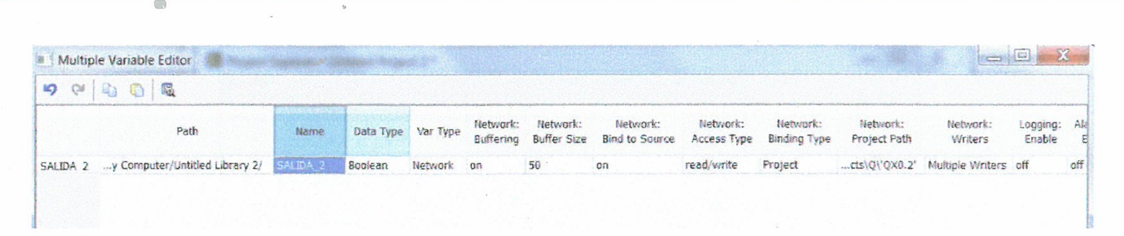


Figura 179

Una vez que se han seleccionado todas las variables, se colocan en el SCADA. Para ello, **se abre un programa nuevo**. En LabVIEW a los programas se los denomina VI («instrumento virtual») (figura 180).

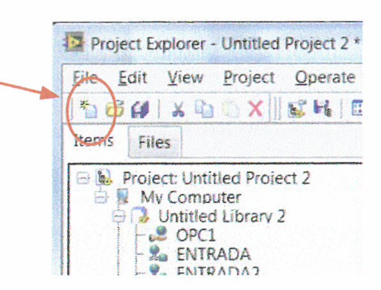


Figura 180

Aquí es donde se abre el VI nuevo (figura 181).

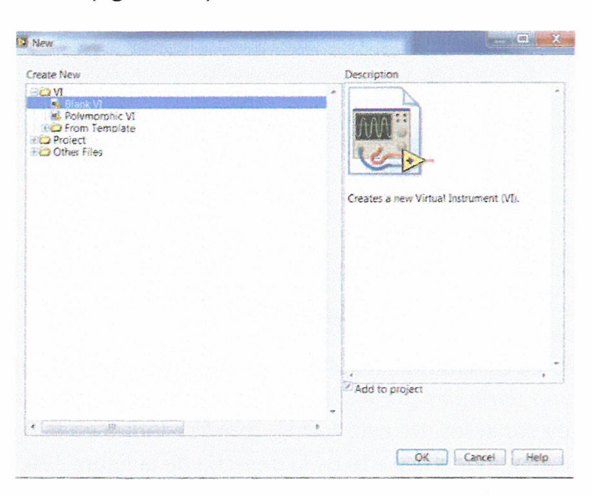


Figura 181

Cuando se abre el LabVIEW, surgen dos ventanas: una se denomina *Bloque de código* y otra *Panel frontal*. En el caso de uso del SCADA, tan solo se utilizará la ventana del panel. Seleccionada dicha ventana, se toca en cualquier parte de su interior y, con el botón derecho, aparecerá el menú de los controles que se pueden colocar. Se selecciona un led y un botón de interruptor, como se refleja en la figura 182.

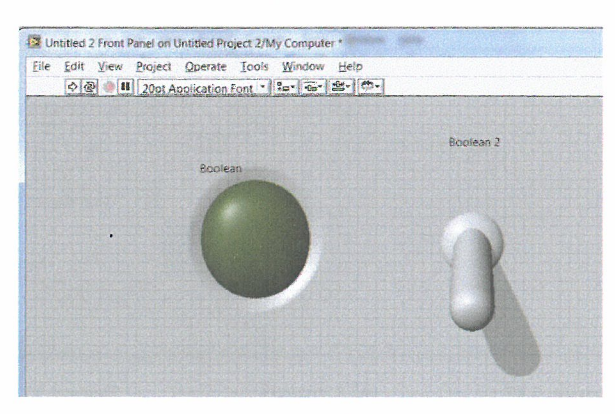


Figura 182

Ahora se deben relacionar esos elementos dispuestos en el panel con las entradas y salidas reales del PLC. Tocando con el botón derecho en el led, y después en el interruptor, se accede a sus propiedades. En este caso, el botón se va a relacionar con la salida del PLC, de forma que el botón del SCADA será el que actuará sobre la salida. Y el led se va a identificar con una entrada del PLC y, así, al activar la entrada del PLC, el led se encenderá.

En las figuras 183-185 se puede ver cómo entrar en las propiedades y cómo se debe seleccionar un tipo de variable, el cual ha de ser del tipo compartida *(Shared Variable).* Luego se debe navegar hasta encontrar la variable deseada; esto se hace desde la pestaña *Data Binding* («enlace de datos»). Las figuras representan la configuración del interruptor. El procedimiento para el led será idéntico.

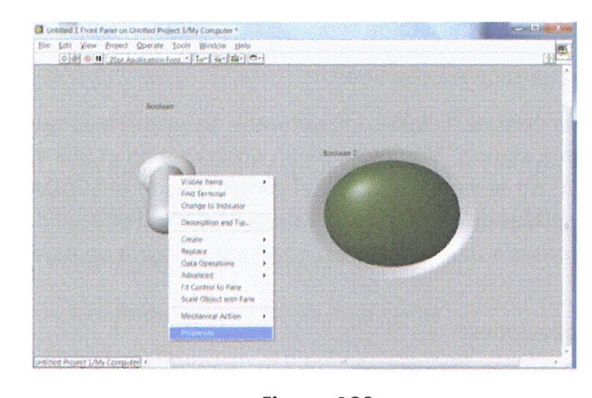


Figura 183

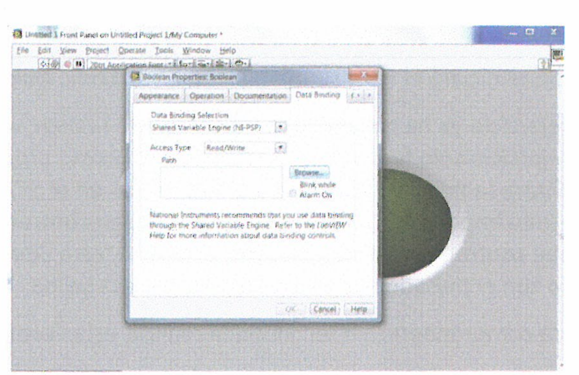


Figura 184

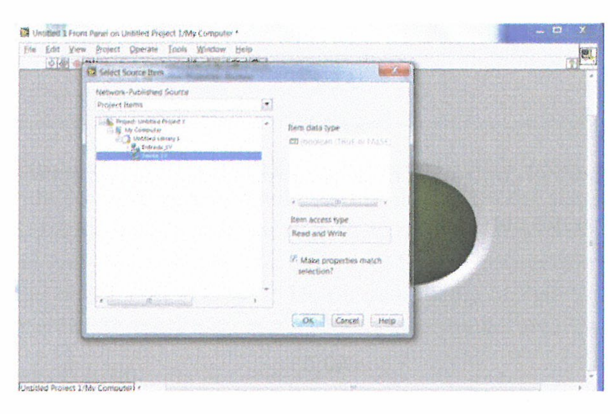


Figura 185

Tras terminar la configuración, se estará en disposición de controlar la salida desde el interruptor y que el led monitorice una entrada del PLC. Para arrancar el SCADA, se acciona el ciclo continuo. Un pequeño triángulo verde al lado de cada elemento del SCADA indica que existe conexión entre el servidor y el cliente.

En la figura 186 se refleja el resultado final.

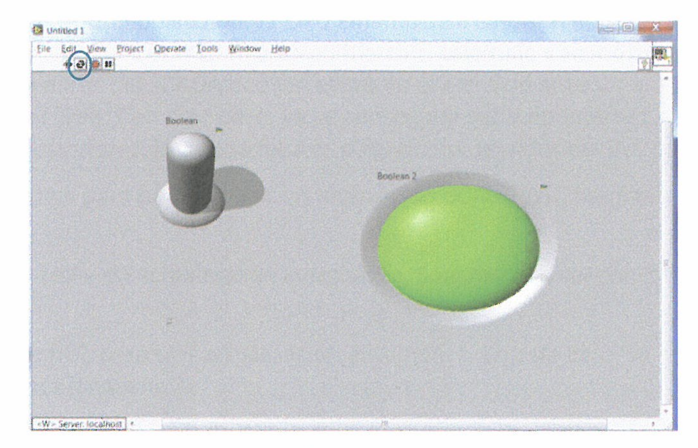


Figura 186

Estos son los puntos que hay que considerar para que el ejercicio no falle:

1. El PLC debe poseer una dirección dentro del mismo grupo que la tarjeta de red ubicada en el ordenador.
2. Se ha de habilitar la opción que permite al PLC usar OPC.
3. Se tiene que contar con las versiones adecuadas para la IE general y OPC.
4. La targeta y el OPC se deben cargar en el *Station Configuration Editor* en el mismo orden que en el proyecto de TIA PORTAL.
5. El nombre de la *Estación PC* del proyecto y el *Station Configuration Editor* ha de ser el mismo.
6. No hay que olvidarse de enviar las configuraciones al PLC y a la tarjeta de red.
7. Se reiniciará el PC si surgiera algún error, como que la conexión no sea *Good*.

El ejercicio ya ha concluido, pero se han de realizar varias consideraciones importantes. La primera es que el ejercicio solo funcionará si en el OB1 no existe programa alguno. Si hubiera algún programa en el que se utilizara la misma salida que a la vez se usa en el SCADA, no se podría realizar el control desde el SCADA. Resulta importante recordar que **no se deben controlar salidas directamente desde el SCADA**. Si una salida debe controlarla el PLC y, a la vez, el SCADA, este último pasaría una marca y esa marca se pondría en paralelo con el pulsador de control del PLC de esa salida.

Para comprenderlo, se recomienda realizar estas comprobaciones:

1. Se hace como en el caso de la práctica, OB1 vacío. Se ha comprobado que no aparecen problemas en el control de la salida A0.0 desde el SCADA porque no se utiliza en el OB1.
2. Se pasa este programa al PLC:

```
U E0.4
=A0.0
```

Se puede comprobar que ahora el control desde el SCADA no funciona correctamente.

3. Se introduce al PLC el siguiente programa:

```
U E0.4
O M0.0
=A0.0
```

Ahora se deberá pasar desde el SCOUT la marca de bit M0.0 y, posteriormente, en el SCADA, se asigna al interruptor esa marca, M0.0.

Se comprobará que todo es correcto, así como que existe control desde el PLC y también desde el SCADA.

Otro aspecto que debe considerarse es que, en este ejercicio, lo único que se ha realizado ha sido la puesta en comunicación del PLC y el SCADA. Un SCADA es mucho más que eso. No solo monitoriza y controla; hay una parte de registro de información y de alarmas realmente importantes en un SCADA. Esto no se va a estudiar aquí. El SCADA de National Instruments DSC, como cualquier otro, también dispone de elementos de registro de históricos y alarmas.

14. Web de usuario con PLC S7-1516

Enunciado

En la era del IoT (Internet de las cosas) no podía faltar, en una temática sobre comunicaciones industriales, en la conexión de los sistemas automatizados, la conexión a Internet. Por ello se va a plantear conectar una CPU 1516 3PN/DP a Internet, con su monitorización y control desde cualquier lugar del mundo. A tal fin, se utilizará la opción de web de usuario del servidor web del PLC.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. SCADA realizado en el ejercicio anterior.
3. *Software* KompoZer para crear páginas web. Vale cualquier otro.
4. Un cable de Profinet para las conexiones.

Realización

Este ejercicio consta de varios pasos y en él se emplean diferentes aplicaciones, alguna ya utilizada en otros ejercicios. Lo que se pretende con este es colgar en Internet un SCADA del sistema de automatización. Ese SCADA se insertará en un servidor de Internet; en este caso, se utilizará el servidor que dispone el SCADA DSC, que ya se empleó en el ejercicio anterior. Por último, el SCADA convertido en página web se enlazará desde la web de usuario que dispone el PLC S7-1516. De esta forma, accediendo con la dirección del PLC desde cualquier parte del mundo, se tendrá acceso a todo el sistema automatizado.

Estos son los pasos para la realización:

1. Desarrollo de un SCADA (sirve el del ejercicio anterior).
2. Configuración del servidor del SCADA DSC *(National Instruments)*.
3. Se diseña una página web para enlazar con el SCADA.
4. Uso del servidor del PLC cargando la página web creada.
5. Configuración del módem para poder subir la web creada a Internet.

Configurar del servidor del SCADA DSC (National Instruments)
En primer lugar, se debe partir del SCADA que se realizó en el ejercicio anterior u otro. Se utilizará el mismo SCADA, DSC y su servidor web. Con este paso, ya se conseguirá subir en Internet el SCADA accediendo con su dirección. En pasos posteriores, lo que se hará es colgarlo del servidor del PLC para facilitar el acceso.

Existen tres procedimientos para poder colgar en Internet un SCADA DSC mediante el servidor de *National Instruments*:

1.º Se monitoriza sin control y sin actualización automática. Solo hay que ver cómo se halla el sistema en ese instante (Monitor).

2.º Se monitoriza sin control, pero con actualización automática. En este caso, se actualiza la página de forma periódica y automática (Snapshot).

3.º Se monitoriza con control (Embedded). Ahora no solo se puede ver, sino que también se puede tomar el control del SCADA. En este caso, resulta obligatorio disponer, en el ordenador remoto, del RUN-TIME nece-

sario para la versión de LabVIEW con el que esté creado el SCADA, o bien disponer del programa de LabVIEW instalado. El Run-Time puede bajarse de la web de National Instruments: www.ni.com.

A continuación, se van a indicar los pasos que se deben seguir para utilizar cada uno de los tres casos. En primer lugar, se habilita el Web Server, un procedimiento común para las tres opciones citadas.

Una vez dentro del programa DSC, y con el SCADA creado anteriormente, desde el menú *Tools* se accede a las opciones (*Options...*). Se describe en la figura 187.

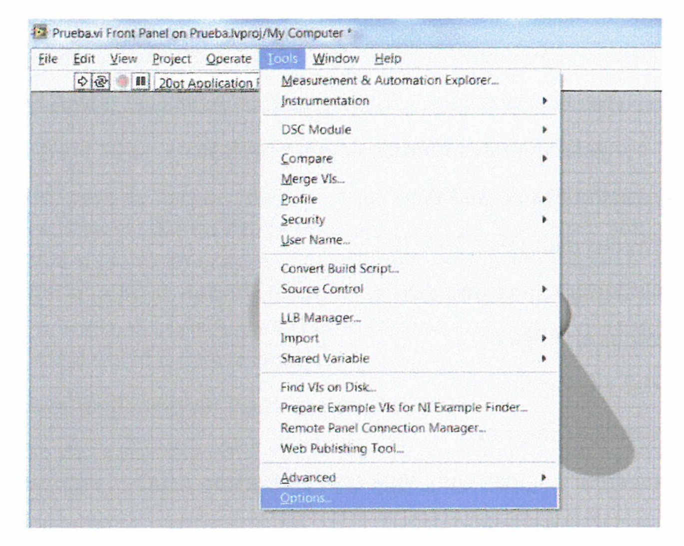


Figura 187

En las opciones, se selecciona *Web Server: Configuration* y se marca la opción de habilitar el Web Server (*Enable Web Server*) (figura 188). Hay que fijarse y apuntar el número de puerto *(HTTP Port)* que indica porque luego es un dato que se deberá conocer.

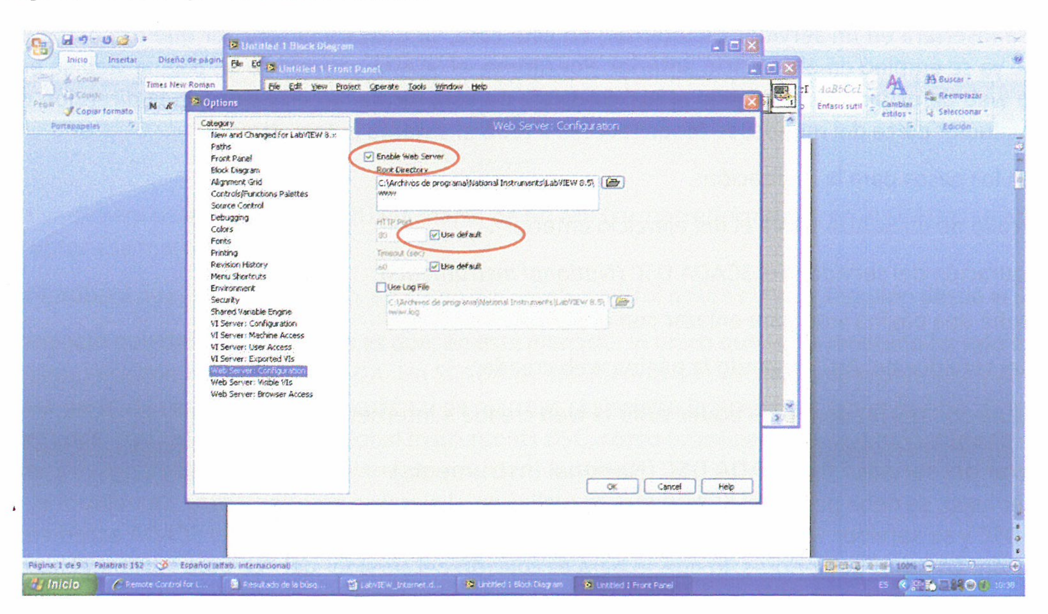


Figura 188

En estos momentos, el servidor de National Instruments ya se encuentra operativo.

Dependiendo de las versiones del LabVIEW, las ventanas pueden cambiar algo, pero siempre se deberá hacer lo que se ha detallado: habilitar el servidor web y observar el puerto por donde sale.

Ahora se accede desde el mismo menú *(Tools)* a la opción *Web Publishing Tool* (figura 189). Allí debe seleccionarse la ruta del VI (el SCADA) y la opción de presentación de la web (sin actualización, con actualización o con control). En la figura 190 se señala cómo hay que proceder.

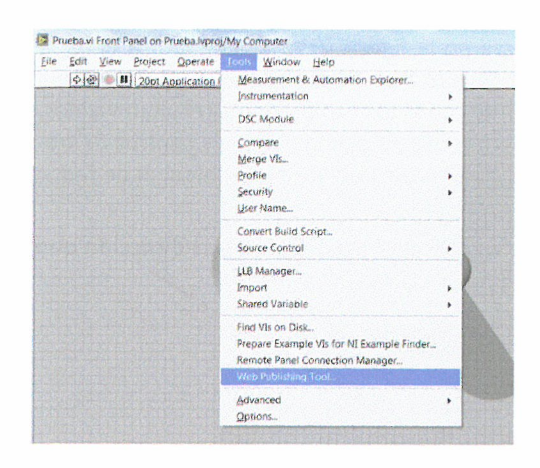


Figura 189

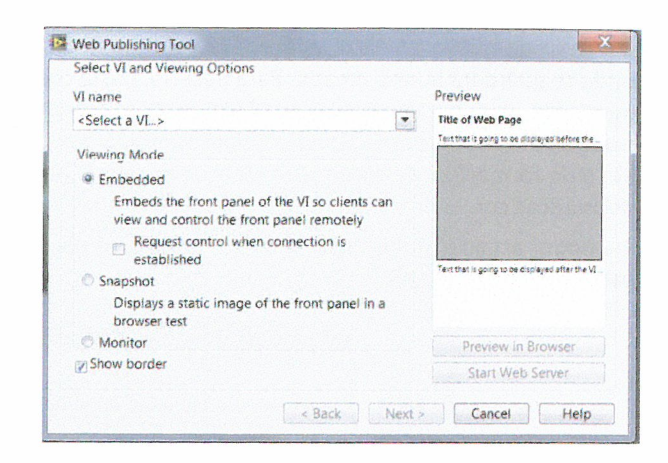


Figura 190

En *VI name* se debe buscar y seleccionar el nombre del SCADA. Luego se escoge el modo. A través del modo *Embedded* se puede controlar. El modo *Snapshot* tiene lugar con una monitorización sin actualización y el modo *Monitor*, con actualización.

Una vez seleccionado el archivo, debe quedar como se representa en la figura 191.

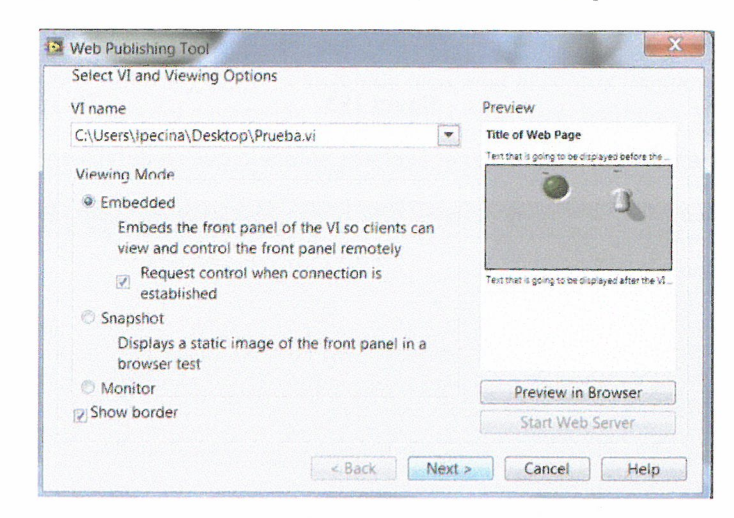


Figura 191

Pulsando en *Next,* se completa el título y otros textos para que aparezca la web (figura 192). Habrá que rellenar los apartados referentes al título de la página, así como al pie y cabecera de páginas, si se desea.

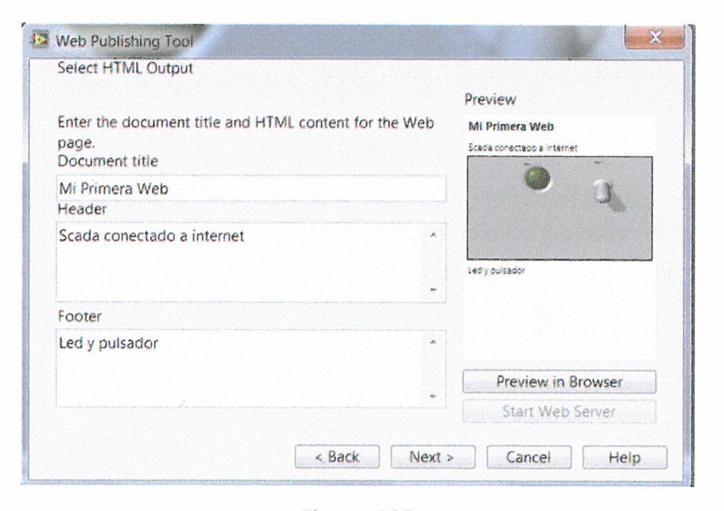


Figura 192

Activando de nuevo en *Next*, aparece la ventana que se observa en la figura 193. Aquí se indica el lugar donde se guardará la web creada. Para hacerlo, basta con pulsar *Save to Disk*. Posteriormente, en la ventana emergente, se conecta para ver en el explorador la web creada. No todos los exploradores resultan válidos. Se debe utilizar Internet Explorer o Safari. También hay que tener en cuenta que, en algunos centros de trabajo o de formación, se bloquea el acceso remoto. Si ocurriese, habría que hablar con el personal de redes informáticas con el fin de liberarlo.

Para poder acceder remotamente a este SCADA, se necesita teclear desde otro ordenador la **dirección** que aparece en la figura 193.

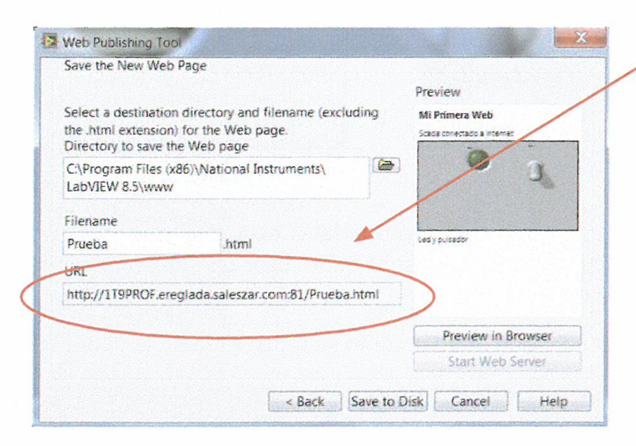


Figura 193

Si se coloca la dirección en el navegador Internet Explorer, saldrá lo que se detalla en la figura 194.

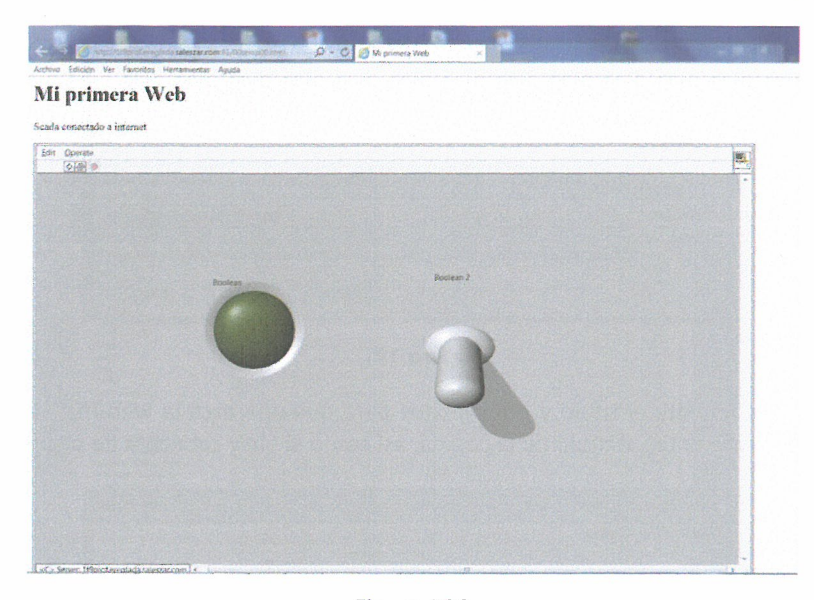


Figura 194

En la opción *Embedded*, se puede tomar el control desde el PLC remoto. En la figura 195 se observa el menú que aparece tocando con el ratón derecho en la pantalla.

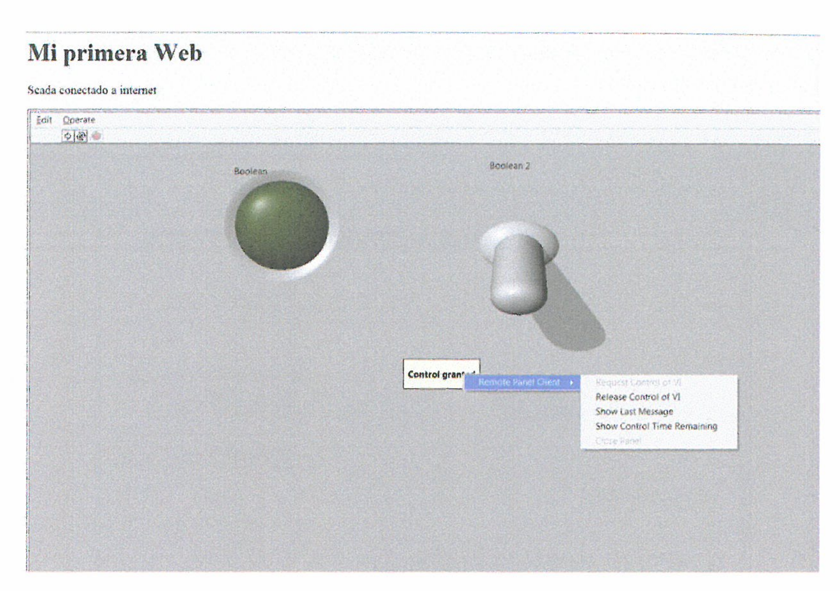


Figura 195

Accediendo a *Remote Control of VI*, se puede tomar el control *(Request Control of VI)* o dejar el control *(Release Control of VI)* del SCADA. Existe un número máximo de participantes que pueden conectarse.

Desde el SCADA (programa LabVIEW), también se puede recuperar el control *(Regain Control)*si se accede de la misma manera que en el caso del navegador, mediante el botón derecho (figura 196).

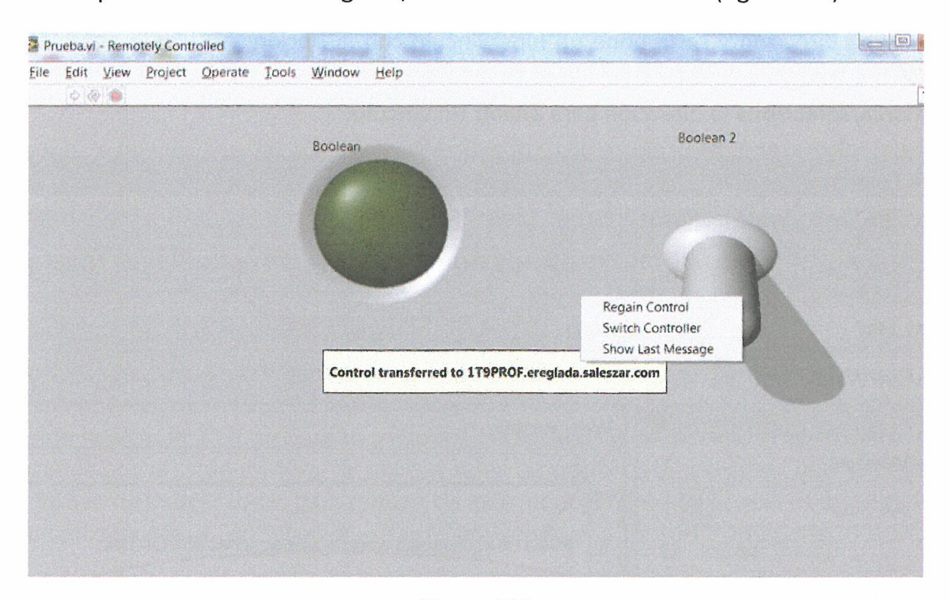


Figura 196

Diseño de una página web para enlazar con el SCADA

Ahora se debe construir una web donde se pondrá, como mínimo, un enlace a la dirección donde se sitúa el SCADA. A esta web se podrá acceder desde el PLC mediante su servidor, como se verá más adelante.

Dependiendo del nivel del que se disponga sobre programación y diseño de páginas web, se puede crear la web de una forma u otra. Aquí se va a utilizar un programa para el que no se necesita conocer nada de HTML. Se trata del KompoZer, un programa gratuito y que se puede bajar del enlace:

http://www.kompozer.net/download.php

Una vez bajado, no hay que instalarlo. Se busca en la carpeta un archivo ejecutable y se abre el programa. La vista inicial es la que aparece en la figura 197.

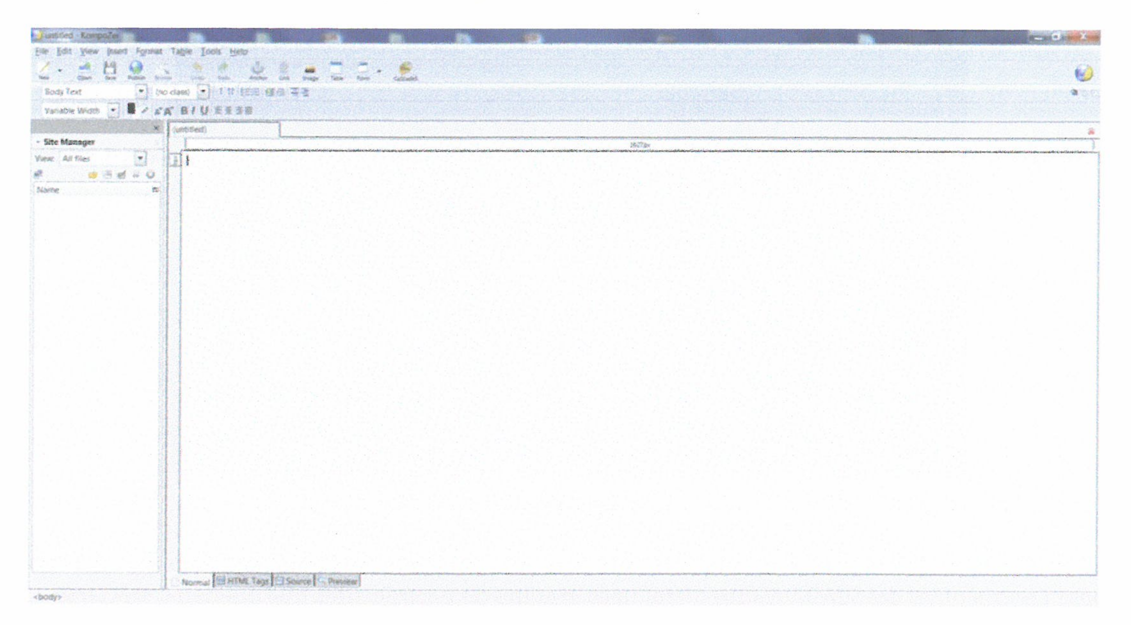


Figura 197

Se escribe un texto explicativo y la dirección que se ha obtenido anteriormente en la página del SCADA. Se selecciona dicha dirección y se le pone un hipervínculo con la opción . En la figura 198 se indica esta situación.

Después, se **marca/selecciona** la dirección para **añadir un vínculo.**

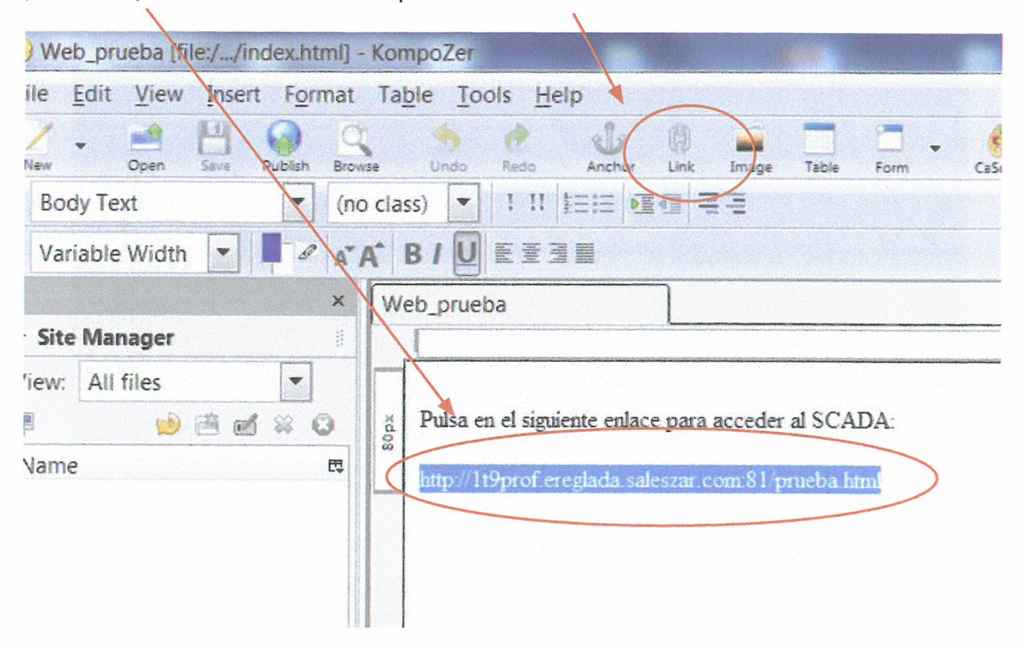


Figura 198

Al accionar el icono de *Link,* aparecen las propiedades, donde tan solo hay que aceptar *(OK),* tal como se aprecia en la figura 199.

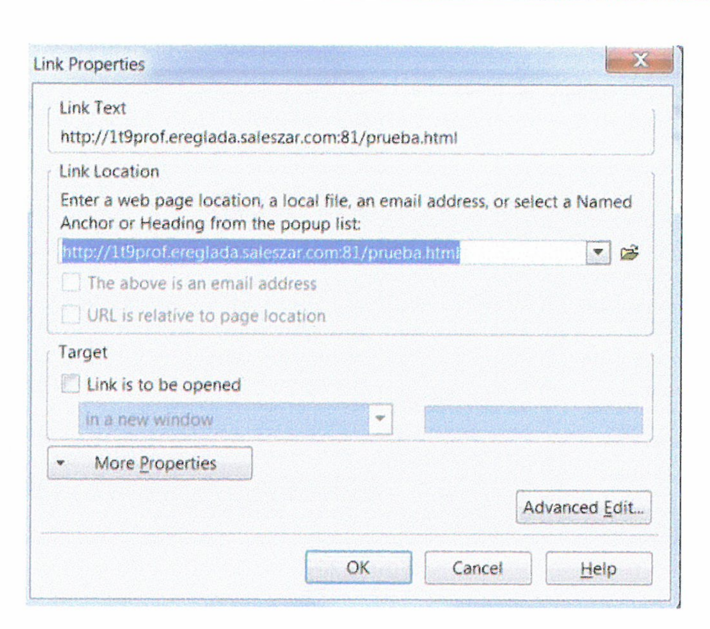


Figura 199

Se guarda el archivo con el nombre de ***Index***, y se le añadirá automáticamente la extensión **html**.

A continuación, se describe cómo añadir esta web al servidor del PLC. Con ello se podrá acceder al SCADA desde la dirección IP del propio PLC.

Webs definidas por el usuario

Este es el último paso para poder acceder al SCADA desde la red local o desde cualquier parte del mundo a través de Internet y por medio del servidor del autómata.

Como ya se sabe, se puede conectar al autómata tecleando su dirección IP en el navegador (preferiblemente Internet Explorer). Desde allí, se accede a la web definida por el usuario, la cual será la web que se acaba de crear con KompoZer.

Los primeros pasos de esta nueva tarea se encaminan a preparar el PLC para que se pueda utilizar su Web Server.

Para ello, se accede a las propiedades del PLC. Desde el apartado *General* se selecciona *Activar servidor web en el módulo*. Después, desde *Actualización automática,* se modifica el tiempo con el que se refrescará la web del propio PLC en el explorador. Por defecto sale 10 s. Por último, se debe generar un usuario y una clave para que, desde la web del PLC, se posean permisos para todas de las opciones. Se escribe un nombre de usuario y, tocando con el ratón en *Nivel de acceso*, salen las opciones de permisos que se desean otorgar a ese usuario. Se han de marcar todas para probar; de este modo, el nivel de acceso es de administrador, que aparece automáticamente. A continuación, se escribe y repite una contraseña (figuras 200-201).

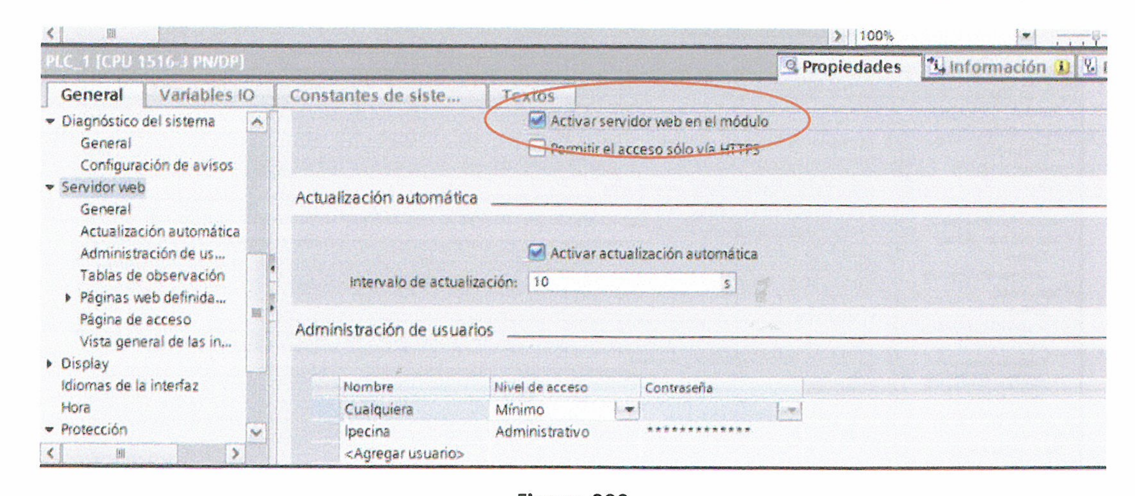


Figura 200

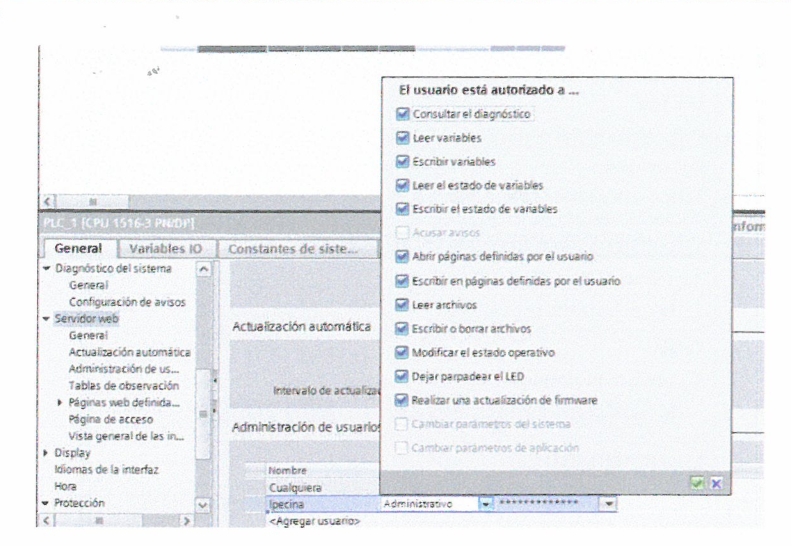


Figura 201

El siguiente paso consiste en colgar la web creada (el SCADA, en este caso) en el servidor del PLC. Para ello se debe acceder, de nuevo, a las propiedades del PLC dentro del apartado *Páginas web definidas por el usuario*. Ahora se trata de completar los apartados indicados en la figura 202. Hay que buscar la web que se ha guardado anteriormente con el nombre de *Index*. Se selecciona el directorio/carpeta donde se encuentra el archivo *(Index) y* se da un nombre a la aplicación; en este caso, se le pondrá SCADA.

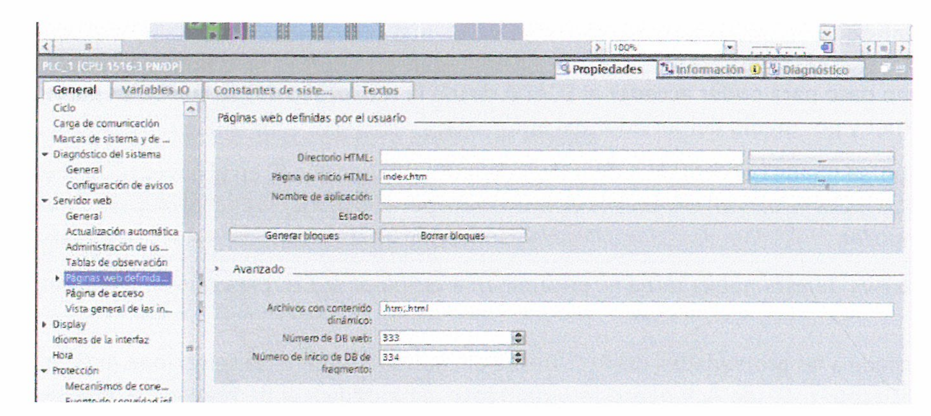


Figura 202

A continuación, deben generarse los bloques (DB333 y DB334) necesarios para cargar la web en el servidor del PLC. Una vez hecho todo esto, el apartado quedará como se observa en la figura 203. Se ha de completar también el tipo de archivo que sale en la *Página de inicio HTML*, poniendo **html y no htm**.

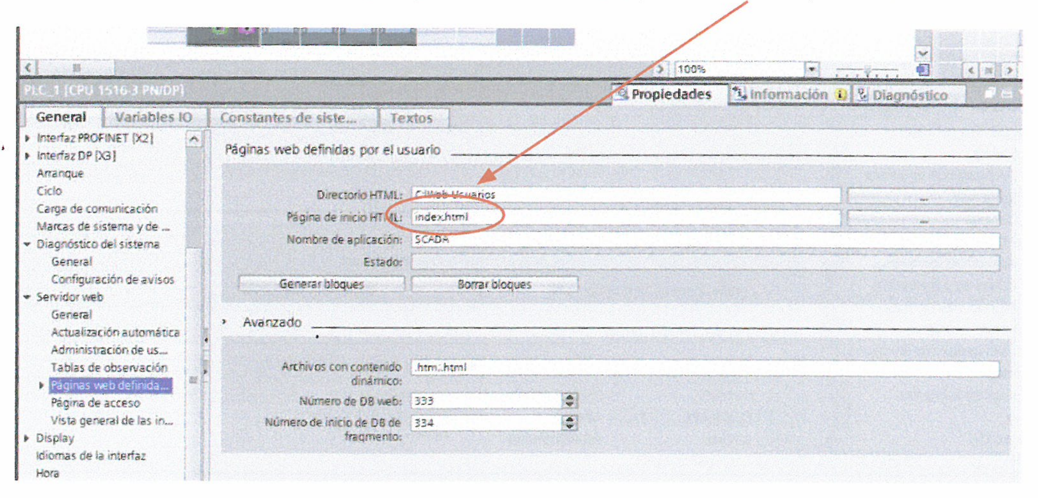


Figura 203

Ahora hay que colocar en el OB1 el bloque que sincroniza páginas web definidas por el usuario. Se abre el OB1 y se busca el bloque *WWW;* después, se arrastra hasta la zona de código (figura 204). En el parámetro *CTRL_DB* se debe colocar el DB333. En SET_VAL, se inserta cualquier marca de palabra.

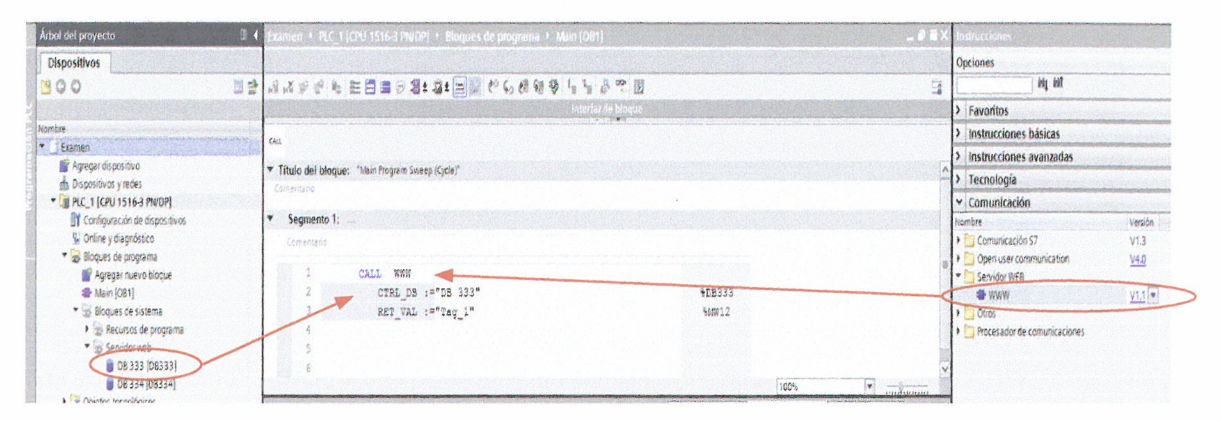


Figura 204

Se carga todo en el PLC y se arranca el navegador Internet Explorer. Es muy importante que el SCADA se halle abierto y funcionando. Se coloca la dirección del PLC en el navegador y se **introduce el usuario y la clave** que se ha dispuesto anteriormente. Véase la figura 205.

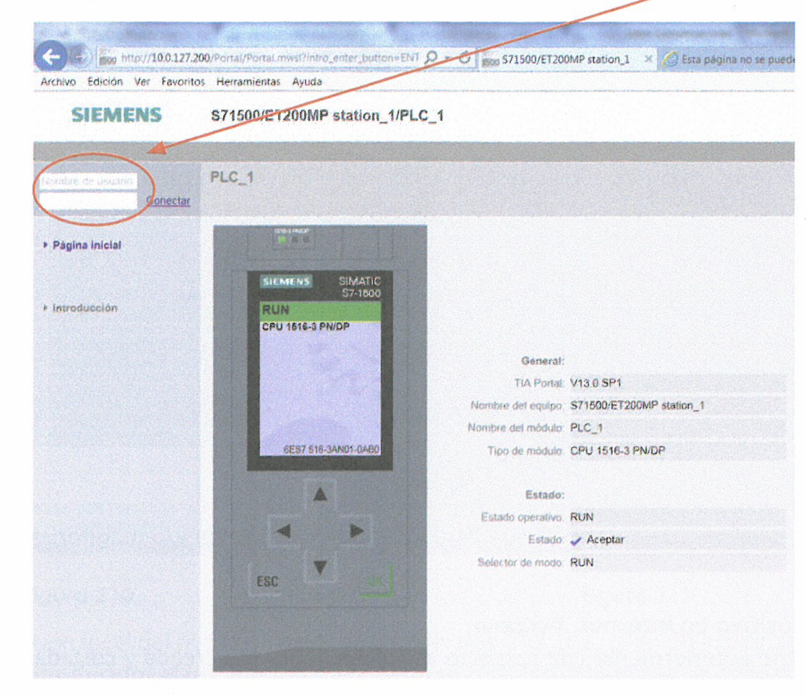


Figura 205

Accediendo a la opción *Página de usuario,* aparece el título que se le ha dado a la aplicación, 0 SCADA; pulsando sobre ella aparecerá la web creada con KompoZer (solo era un enlace al SCADA). Véanse figuras 206 y 207.

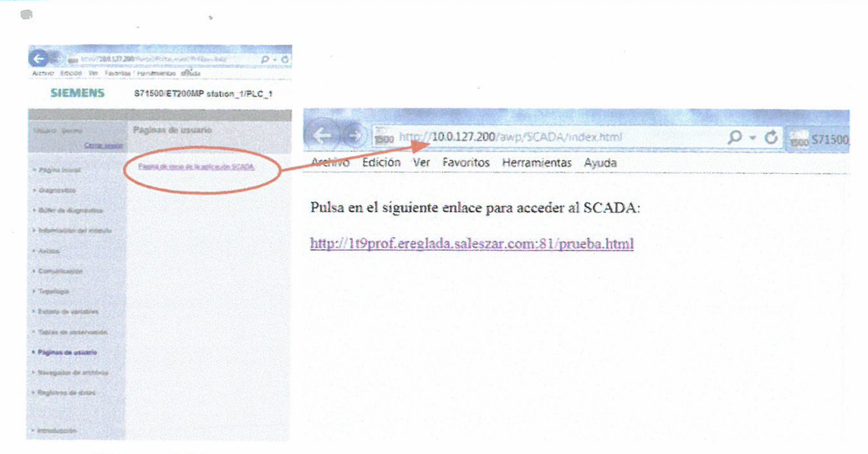


Figura 206 Figura 207

Al pulsar en el enlace que se puso en la web creada con KompoZer, se accede al SCADA, como se aprecia en la figura 208.

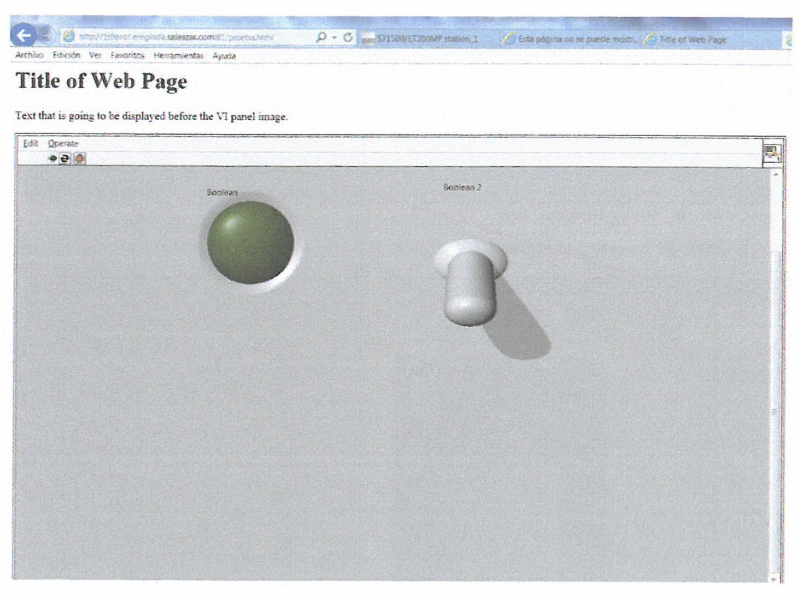


Figura 208

Desde el navegador y desde cualquier parte del mundo puede controlarse y/o monitorizar el proceso de la automatización.

Subida de un dispositivo en Internet. Accceso
Todo lo dicho y hecho anteriormente con respecto al acceso de la web creada y colgada en el servidor de National Instruments, así como la dirección del PLC, solo sirve para acceso local en la red privada donde se ha creado. Para poder colgar un dispositivo en Internet, sea el PLC, una WEBCAM IP, una aplicación como la del SCADA, etc., se debe dejar paso a la gran red de redes, Internet, a través del módem. Lo primero que se necesita es conocer la dirección IP local (privada) del dispositivo, la dirección de la tarjeta del ordenador donde se sitúa el SCADA.

También hay que escribir la **dirección privada del módem o *router*** por donde va a salir el autómata. Se encuentra en las propiedades del PLC en la opción *Direcciones Ethernet*, en *Protocolo IP* (figura 209).

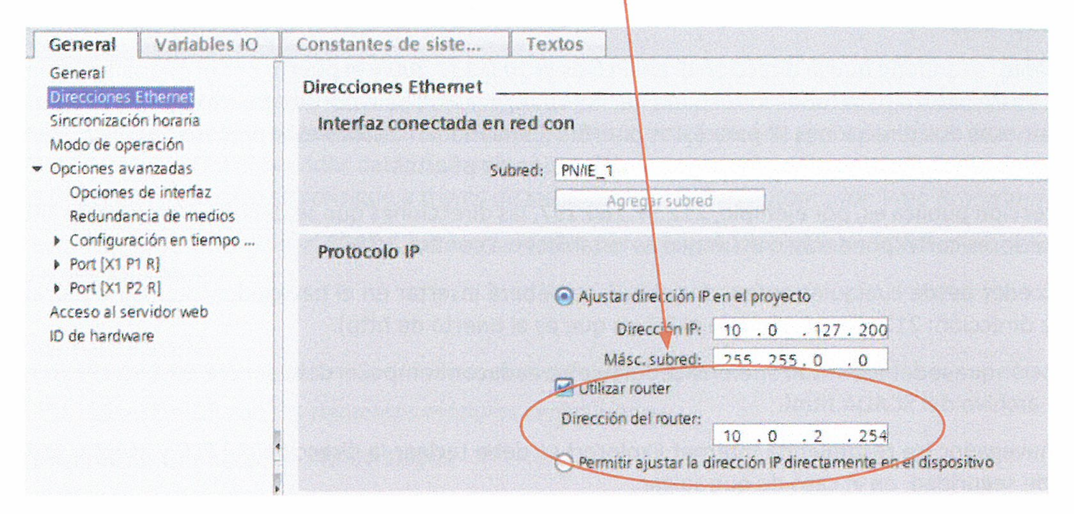


Figura 209

Para saber la dirección de la tarjeta de red del ordenador, se ha de ejecutar el comando «cmd» en *Inicio*. Esto dependerá del sistema operativo que se disponga. Una vez ejecutado el comando «cmd», se abrirá la consola *MsDos* y allí se teclea el comando «IPconfig», de modo que aparece la dirección IP del ordenador ***(IPv4)***, como se ve en las figuras 210 y 211.

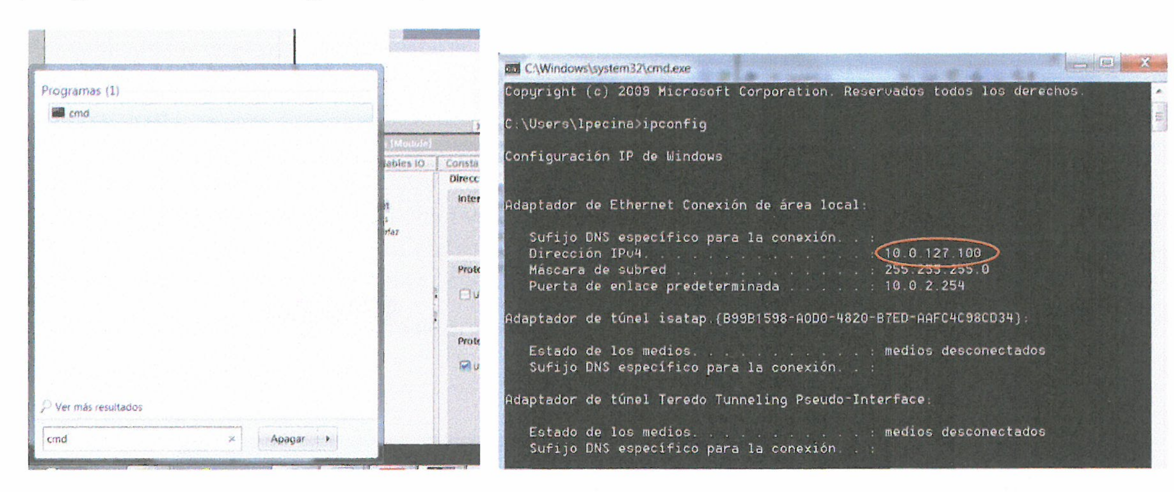


Figura 210 Figura 211

Tecleando la dirección local del punto de acceso (será del tipo 192.168...), en un explorador, se entrará en el módem, una vez introducidos el usuario y la clave. Si no se conocen esos datos, se deberá buscar en Internet los valores por defecto para ese módem en particular.

Una vez dentro del módem, se tendrá que dar acceso a esa dirección IP privada a través del puerto adecuado. Para ello, se necesita saber el puerto por donde sale la aplicación. Si se desconoce el método de abrir los puertos en un módem concreto, habrá que investigar sobre el método en Internet; por ejemplo, si nuestro PLC tiene dirección 192.168.1.20 y el puerto por donde sale es el 80, se deberá abrir el puerto 80 para esa dirección (abierto por defecto porque es por donde salen las páginas web), tanto para TCP como UDP, en el caso de que se diese la opción de elegir. No se olvide de guardar los cambios en el módem.

Ahora, desde cualquier navegador, debe escribir la dirección pública con la que sale el módem al exterior, separando el puerto por el que sale con dos puntos (:) (p. ej., 212.97.161.187:80). Para conocer la dirección pública, se puede utilizar cualquier web que nos la indique, como:

http://cual-es-mi-ip-publica.com/

Para este ejercicio, se va a suponer que la tarjeta de red del ordenador posee la dirección IP privada 10.0.127.100 y que el puerto que se ha seleccionado en la aplicación LabVIEW del SCADA, según se ha visto, era el puerto 81.

Igualmente, se supone que la dirección IP del PLC es la 10.0.127.200 y el puerto por el que sale por defecto, el 80 (es este puerto para todos los PLC; de hecho, cualquier web sale por ese puerto). En el módem habrá que abrir esas dos direcciones IP para estos puertos. Cada módem realiza este proceso de una manera distinta: puede venir como puertos de juegos o redirección de puertos.

Si la dirección pública es, por ejemplo, 212.97.160.197, las direcciones que se deben poner para las situaciones anteriores corresponderán con las que se establecen a continuación.

Para acceder desde cualquier ordenador al PLC, se deberá insertar en el navegador Internet Explorer la siguiente dirección: 212.97.160.197 (sin el 80, ya que es el puerto de http).

La dirección que se deberá incluir en el enlace de la web creada con KompoZer debe ser 212.97.160.197:81/,nombre del archivo del SCADA.html.

En un navegador (se recomienda Internet Explorer) se debe teclear la dirección del PLC y aceptar todos los avisos de seguridad, en el caso de que salgan.

Puede salir una pantalla de aviso de peligro como la de la figura 212. Se deberá acceder a *Opciones avanzadas*. A continuación, saldrá la pantalla de la figura 213 y deberá accederse a la dirección indicada.

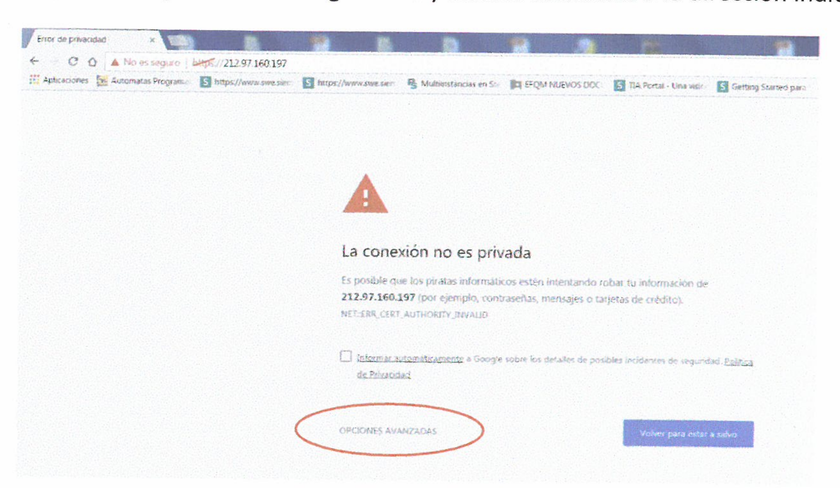


Figura 212

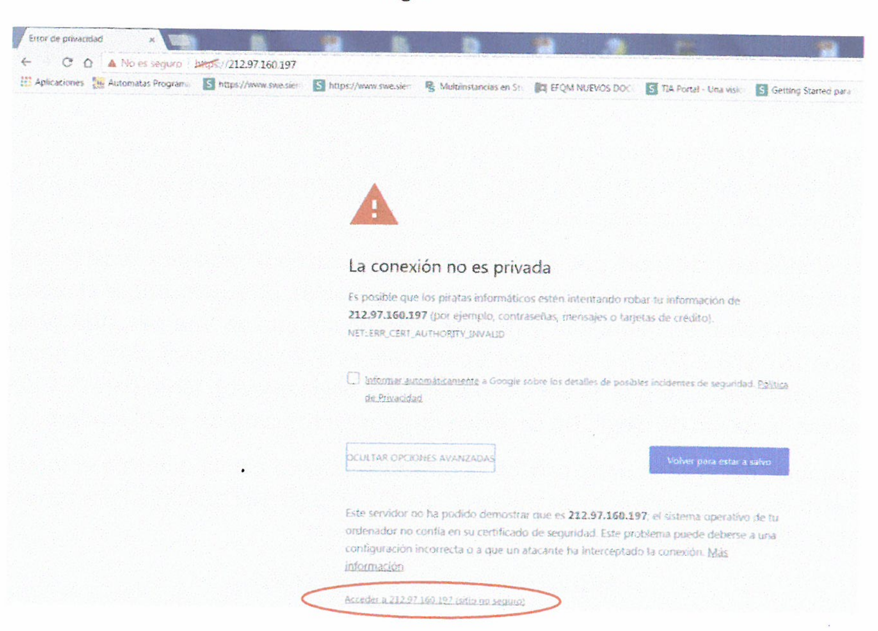


Figura 213

15. Web de usuario con PLC S7-1516 utilizando comandos AWP

Enunciado

Lo realizado en el ejercicio anterior es un ejemplo de lo que se puede hacer utilizando una web que corre en otro servidor externo del PLC, que se enlaza con el del PLC. Existe otra forma de proceder: crear una web que directamente se cuelga en el servidor del PLC. Para que sea efectiva esa web, debería poder interactuar con variables del propio PLC. Esto se consigue a través de los comandos AWP *(Automation Web Programming)*.

En este ejercicio se va a sustituir el SCADA del caso anterior por una web diseñada en HTML a la que se han añadido comandos AWP.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. *Software* KompoZer para crear páginas web. Vale cualquier otro.
3. Un cable de Profinet para las conexiones.

Realización

Como se ha destacado en el enunciado, los pasos del SCADA y la conexión al servidor de National Instruments no serán necesarios ahora. El objetivo estriba en diseñar una web en HTML utilizando los comandos AWP. Será necesario conocer dichos comandos y la manera de utilizarlos en el editor de páginas web. Se utilizará de nuevo el *software* KompoZer como editor HTML.

Con los comandos AWP, se pueden realizar las siguientes acciones:

- Lectura de variables del PLC.
- Escritura de variables del PLC.
- Lectura de variables especiales.
- Escritura de variables especiales.
- Definición de tipos Enum.
- Asignación de tipos Enum a variables.
- Definición de fragmentos de bloques de datos.
- Importación de fragmentos de bloques de datos.

La sintaxis de los comandos AWP (excepto para lectura) tiene esta estructura:

<!-- AWP_<nombre de comando y parámetro> -->

Resulta importante que el juego de caracteres que se utilice en la creación de la web sea UTF-8 (8-bit *Unicode Transformation Format*). Para ello se deberá incluir la siguiente línea en el editor web:

<meta http-equiv="content-type" content="text/html; charset utf-8">

La sintaxis para los comandos AWP más comunes se pueden observar en la siguiente tabla:

Función	Representación
Lectura de variables PLC	:=<Varname>:
Escritura de variables PLC	<!-- AWP_In_Variable Name='<Varname1>' -->
Lectura de variables especiales	<!-- AWP_Out_Variable Name='<Typ>:<Name>' -->
Escritura de variables especiales	<!-- AWP_In_Variable Name='<Typ>:<Name>' -->
Definición de tipos Enum	<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='0: "<Text_1>",1:"<Text_2>",...,x:"<Text_y>"' -->
Asignación de tipos Enum a variables	<!-- AWP_Enum_Ref Name='<Varname>' Enum='<Name Enum-Typ>' -->
Definición de fragmentos de bloques de datos	<!--AWP_Start_Fragment Name='<Name>'[Type=<Typ>] [ID=<Id>] -->
Importación de fragmentos de bloques de datos	<!--AWP_Import_Fragment Name='<Name>' -->

Para poder interactuar con una variable del PLC desde la web creada, es necesario que previamente se hayan creado símbolos/nombres para cada una de ellas.

En este libro no se va a aprender a crear una página web html ni en JavaScript. Tan solo se describirá de qué forma se puede enlazar una variable entre el PLC y la página web creada, ya sea para leerla desde el PLC o escribir en el PLC. Para quien desee ampliar conocimientos de creación de páginas web, existe bastante documentación que enseña cómo hacerlo de forma clara. En el siguiente enlace de Siemens se puede aprender más sobre el tema:

https://cache.industry.siemens.com/dl/files/496/68011496/att_14832/v1/68011496_html_basics_for_simatic_cpus_en.pdf

Igualmente puede hallarse información en el libro *Informática Industrial* de esta misma editorial, en la colección Marcombo Formación.

Lectura del PLC en una web

En este apartado se visualizará el estado de una variable del PLC. Esta variable será un contador (Z1) que se incrementa de forma continua cada segundo utilizando una marca de ciclo. En el OB1 se incluirá:

```
U  Marca_ciclo_1seg    //Bit 4 de la marca de ciclo
ZV Contador            // Contador  Z1
```

En KompoZer se añadirá el código mostrado en la figura 214 en la carpeta ***Source***, que es el editor de html.

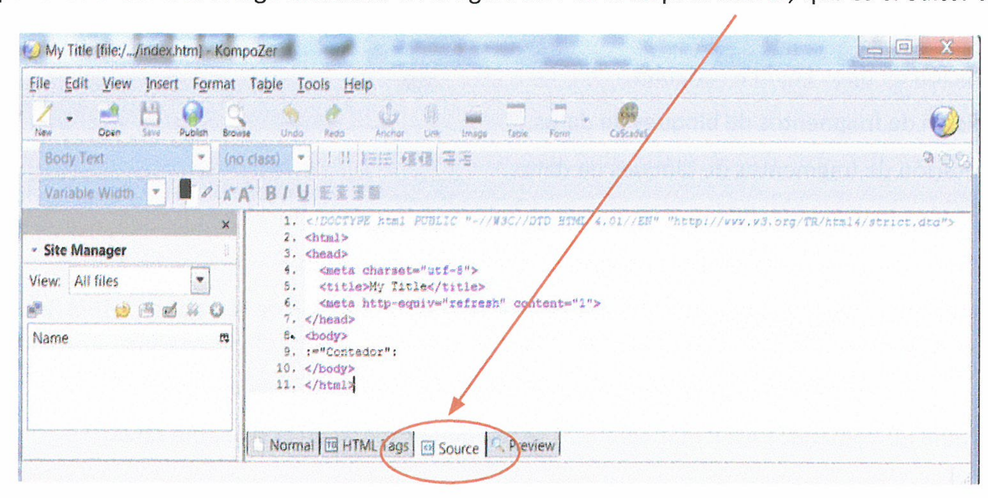


Figura 214

Si no se tienen conocimientos de html, resulta difícil interpretar el significado del archivo. Para aclarar su significado, se comentará cada paso.

En un archivo html, existe una cabecera *(head)* donde se colocan aquellas características generales del resto del archivo; por ejemplo, la línea *<meta charset="utf-8">* es el tipo de codificación del texto que, para estas aplicaciones, debe ser obligatoriamente UTF-8. Con la línea *<meta http-equiv="refresh" content="1">* se lleva a cabo un refresco automático cada segundo. Cambiando el número, se modifica el tiempo de refresco. Se ha de aclarar que este tiempo de refresco se corresponde a la página web. El PLC también dispone de un tiempo de refresco, pero ese hace referencia a su servidor web.

Otra sección es el cuerpo del archivo *(body)*. Aquí se coloca el grueso del archivo. En este caso, tan solo se sitúa el nombre de la variable que se va a leer del PLC, según el formato para los parámetros AWP y visto anteriormente.

Este archivo se debe guardar con el nombre obligatorio de **index.html**. Seguidamente, se inserta este archivo en el servidor del autómata. Se va a proceder a repasar los pasos que, en parte, ya se hicieron anteriormente. Se supone que el autómata tiene activada la opción de servidor web y que se han dado derechos para poder acceder a la web de usuario del PLC. Este proceso se describe en las figuras 200 y 201 del ejercicio anterior.

Seguidamente, se cuelga la web creada con el KompoZer en el PLC y se coloca en el OB1 del PLC el bloque de función WWW. Puede verse en las figuras 204 y 205 del ejercicio anterior.

Una vez realizado, se observa y se compara una tabla de observación de variables del PLC con la web en el navegador (figura 215).

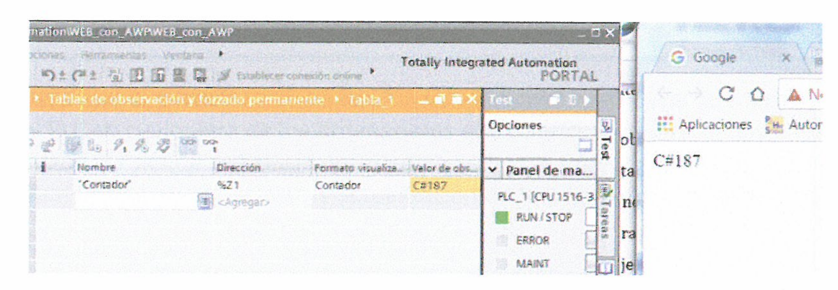


Figura 215

Escritura en una variable del PLC

Al ejemplo anterior se le va a añadir la posibilidad de escribir un dato en un registro del PLC. El procedimiento será el mismo que en el caso anterior; solo se deberá añadir en el archivo html la parte correspondiente a la escritura en el PLC.

Se crea, en la tabla de datos del PLC, una marca de palabra. En este caso, a la dirección MW10 se le da el nombre de *Marca1*.

El archivo de html será el mostrado en la figura 216.

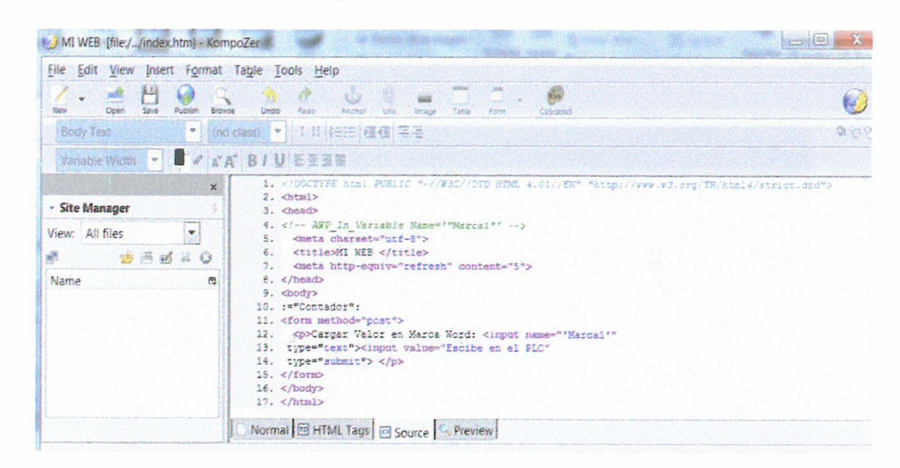
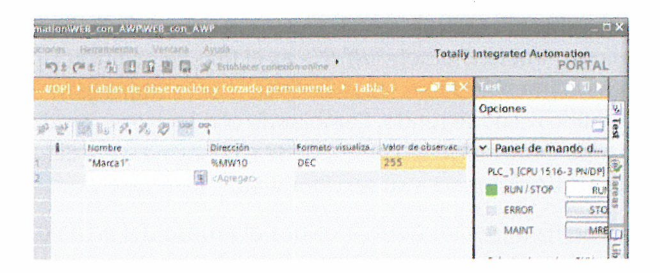


Figura 216

Se puede ver en la tabla de observación de variables del PLC cómo, al accionar el botón *Escribe en el PLC*, pasa el valor introducido a la marca MW10 (como se representa en la figura 217).



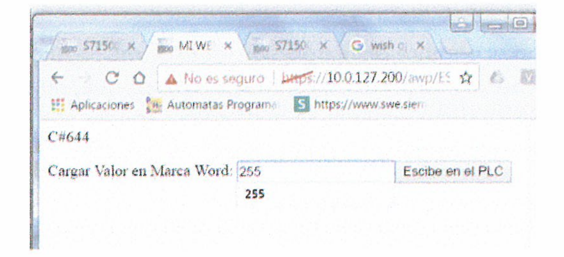


Figura 217

16. Página web de una CPU 1500

Enunciado

Una de las ventajas de utilizar Profinet es que, mediante esta comunicación, se puede salir de forma rápida por Internet o por la red local.

En este ejercicio se va a tratar de visualizar la información que se puede tener de una CPU de forma remota, ya sea localmente o a través de Internet.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
   - Un cable de Profinet para las conexiones.

Realización

En primer lugar, se activa el servidor web de la CPU. Una vez hecho, solo hay que teclear en un navegador la dirección IP del PLC en cualquiera de los ordenadores que se encuentren en la misma red que el PLC. En la figura 218 se puede ver dónde se halla la opción para activar el servidor web. Una vez marcada la opción, se envía al PLC. También se puede activar la actualización de la página con un tiempo determinado de actualización.

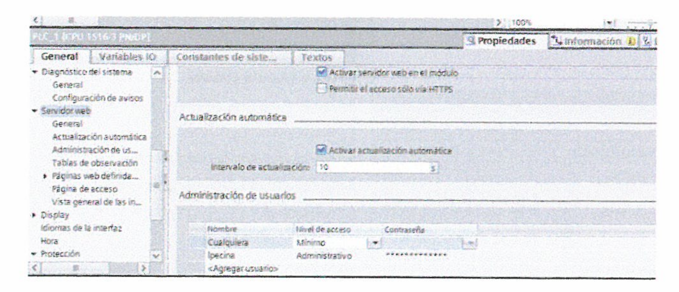


Figura 218

Tecleando la dirección IP del PLC en un navegador, aparecerá la página web del PLC. Presenta la vista que aparece en la figura 219.

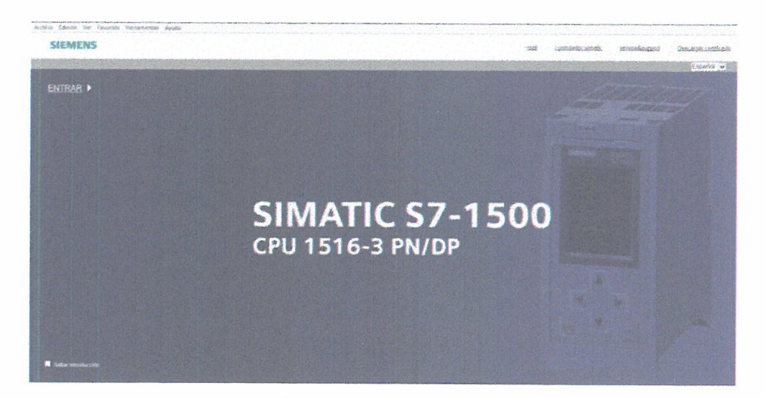


Figura 219

Pulsando en ENTRAR, se accede al contenido de la web. Si no se ha creado perfil con derechos, solo se podrá ver lo que aparece en la imagen de la figura 220 que, como se puede apreciar, constituye muy poca información. Para poder acceder a más, se debe crear un usuario con clave y asignarle derechos de acceso.

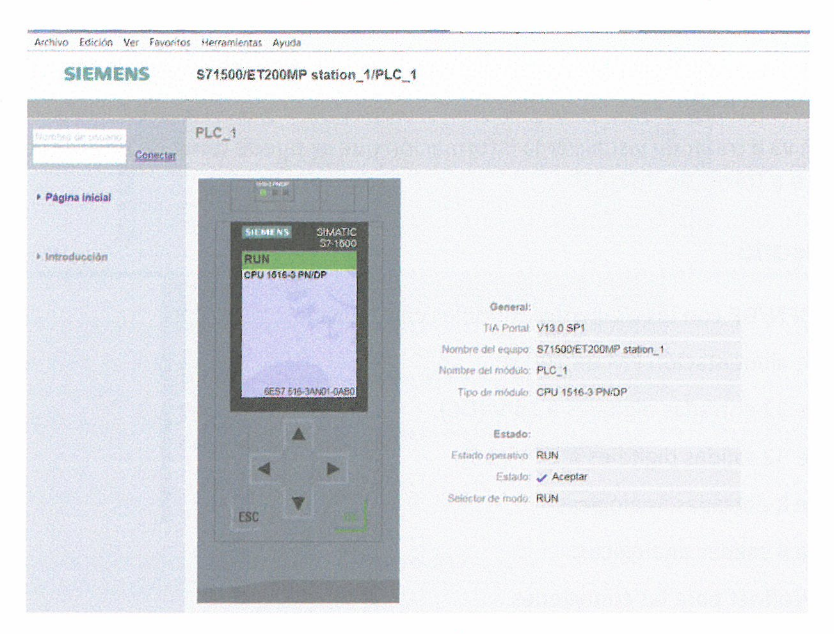


Figura 220

Los usuarios se pueden crear en las propiedades del PLC en la sección de web, como se aprecia en la figura 221. Si se marcan todas las opciones posibles, el usuario que se crea automáticamente es el de Administrador.

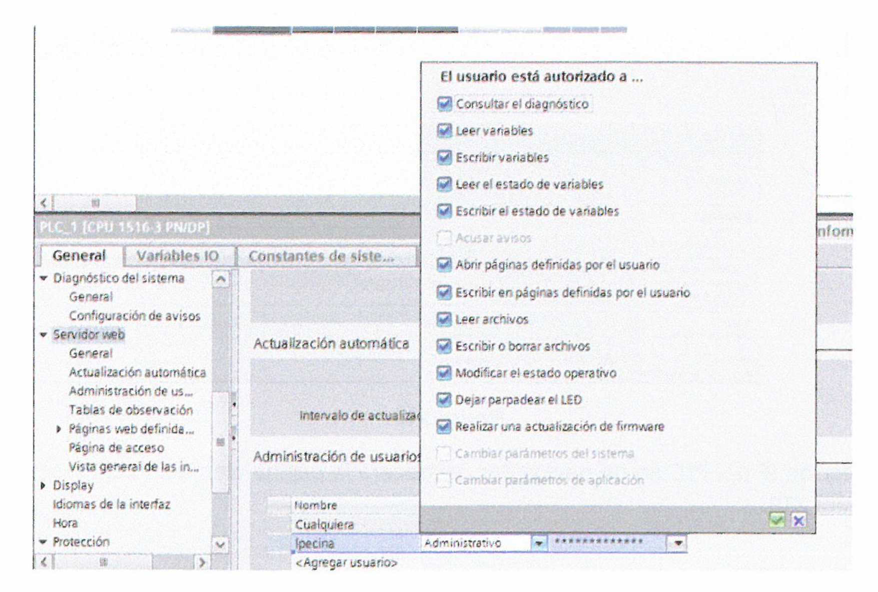


Figura 221

Una vez creada la cuenta de usuario y entrando con esos datos, se accederá a todas las posibilidades que ofrece la web (figura 222).



Figura 222

Página inicial

Desde esta página se pueden observar las características más sobresalientes del PLC. También se utiliza parar, arrancar y poner intermitente el led del PLC.

Diagnóstico

En esta página web se accede a la siguiente información (dependiendo de la CPU):

- Identificación.
- Protección del programa.
- Memoria.
- Tiempos de ejecución.

Búfer de diagnóstico

Disponer del búfer de diagnóstico de forma remota resulta muy importante. También se puede guardar este documento en formato CSV. En la figura 223 se indica cómo **guardar** el búfer de diagnóstico.

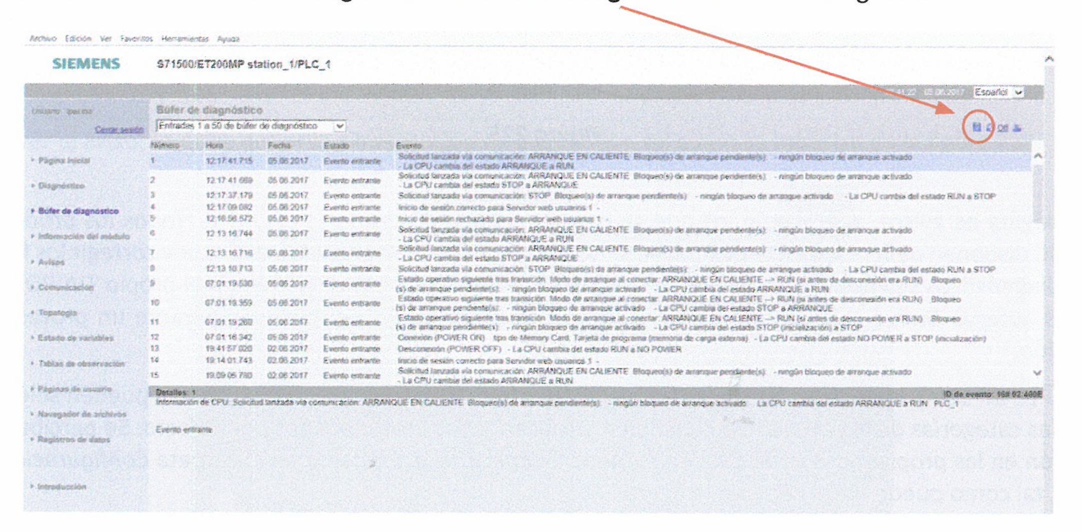


Figura 223

Información del módulo

En esta página se informa del estado de los módulos que son accesibles. Se puede navegar entre los diferentes módulos y submódulos indicando su estado de funcionamiento. Si se dispone de una red de Profinet, también se puede acceder a los diferentes módulos de los *Devices* conectados. Desde aquí también se actualiza el *firmware* de los módulos, tal como se distingue en las figuras 224 y 225.

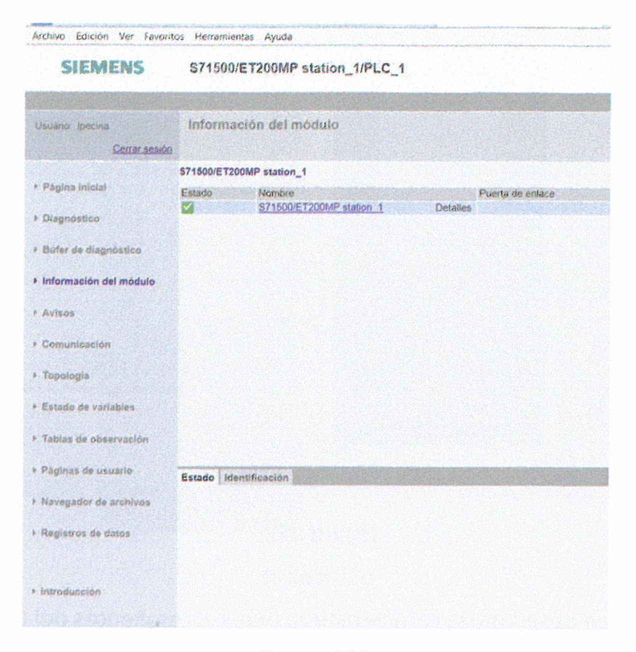


Figura 224

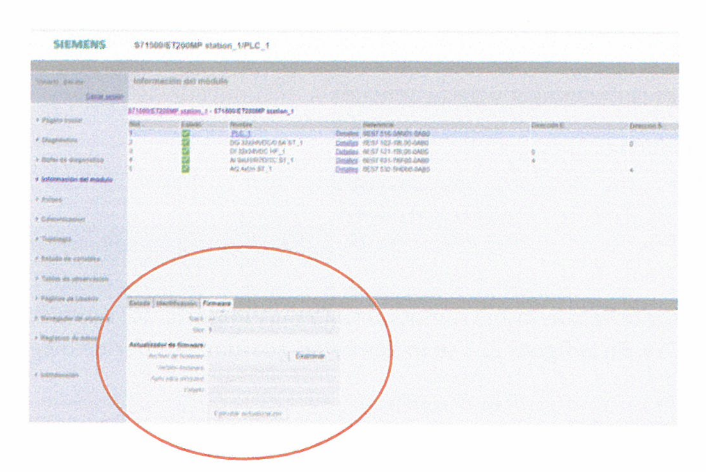


Figura 225

Avisos

En la página de avisos, aparecerán los que se hayan configurado desde TIA PORTAL. Todos los productos SIMATIC disponen de funciones de diagnóstico integradas con las que se puede detectar y corregir los fallos. Estos diagnósticos se visualizan en el *display* del PLC S7-1500 o vía web. También en el propio TIA PORTAL, con los accesos *Online* y *diagnóstico*, se visualizan y analizan los fallos ocurridos durante un proceso en ejecución.

En los PLC S7-1500, la opción de diagnóstico del sistema siempre se encuentra activada y pueden seleccionarse las categorías de los avisos. Estos también se hallan todos seleccionados por defecto. Se percibe esta situación en las propiedades de la CPU en el menú *Diagnóstico del sistema* en la carpeta *Configuración de avisos*, tal como puede observarse en la figura 226.

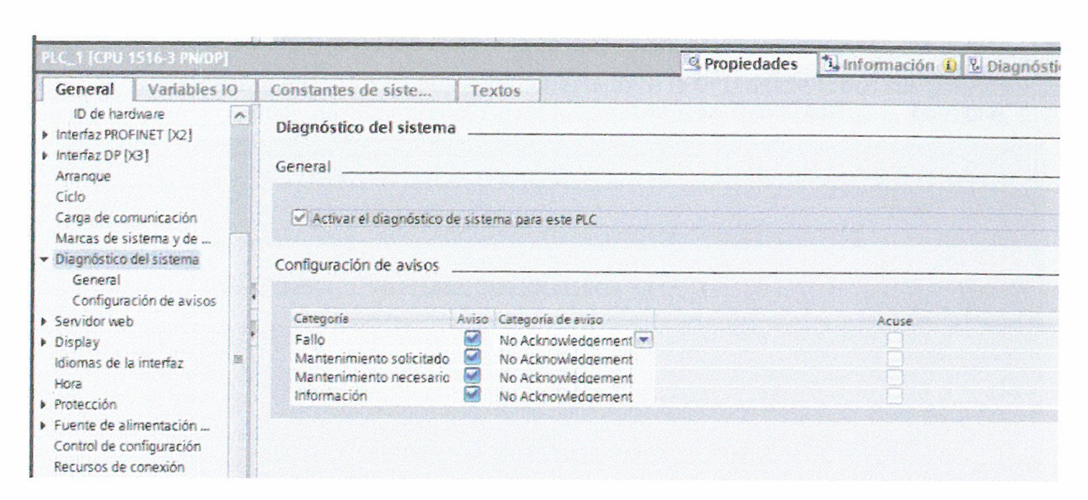


Figura 226

Como se ha dicho, cada uno de los dispositivos posee una configuración de avisos. Se va a poner un ejemplo para que se active un aviso y pueda ser visto desde la web en el apartado *Avisos*.

En la tarjeta de salidas analógicas se desea que aparezca un aviso de error cuando un canal de esa tarjeta sufra un fallo de alimentación. Para configurar dicho aviso, se deberá acceder a la tarjeta desde TIA PORTAL y seleccionar ese aviso de error. Se puede ver en la figura 227 cómo acceder al diagnóstico del canal 0 de esa tarjeta de salidas analógicas. Una vez seleccionado el aviso se cargarán en el PLC los cambios realizados.

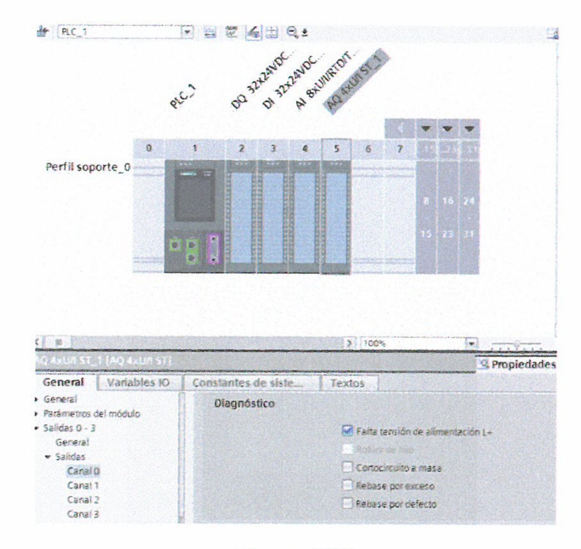


Figura 227

Para ver el aviso, se debe quitar la alimentación de esa tarjeta, tal como se muestra en la figura 228.



Figura 228

En la web, dentro del apartado *Avisos*, se indica cuál es el motivo del error (figura 229).

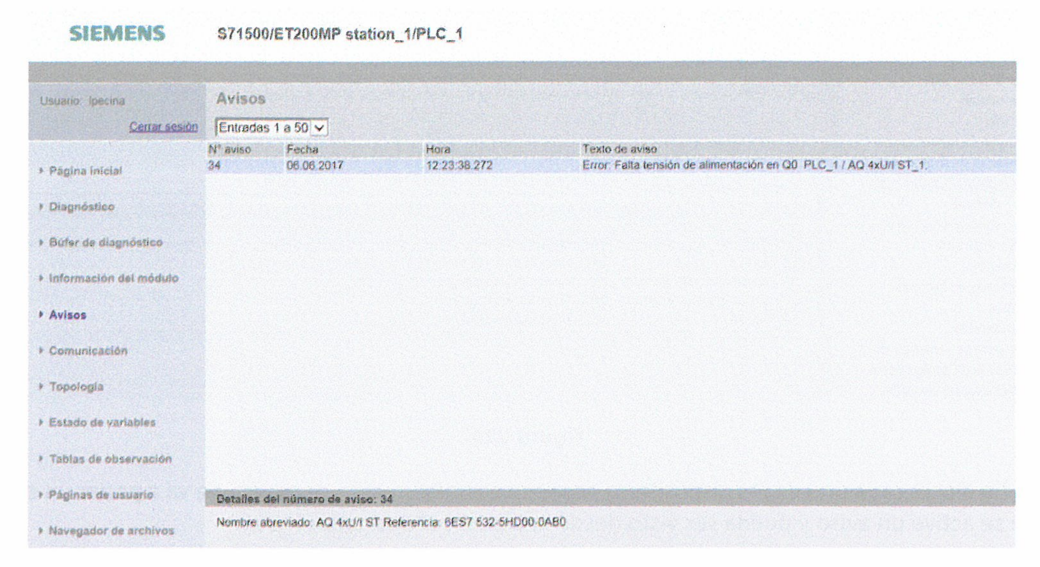


Figura 229

Estos avisos también se observan en el *display* del PLC S7-1500 y desde la opción *Online y diagnóstico* de TIA PORTAL.

En la figura 230 se aprecia el aviso en el *display* del propio PLC. Aparece en la carpeta *Avisos* del menú *Diagnóstico*.



Figura 230

Comunicación
En esta página se observan datos sobre las comunicaciones y su configuración.

Topología
En esta sección se visualizan todos los dispositivos conectados a la CPU desde todos sus puertos. Indica los datos de direcciones y también permite acceder a los posibles errores que exista en la red. En la figura 231 se indica la topología de una red de Profinet sin alteraciones. En dicha red, se encuentran un PLC S7-1516 y una ET 200SP. En la figura 232 se representa la misma instalación con el cable de Profinet cortado, que une el PLC a la ET. Se puede comprobar que notifica error.

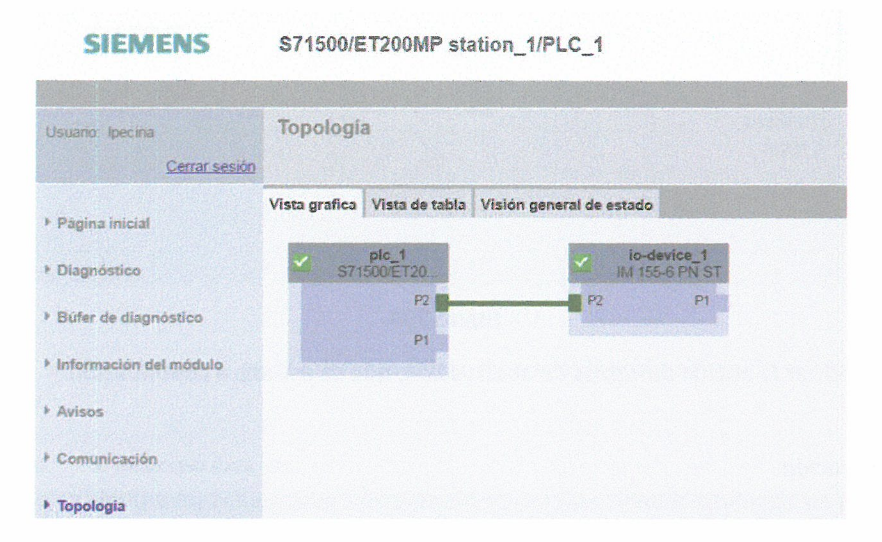


Figura 231

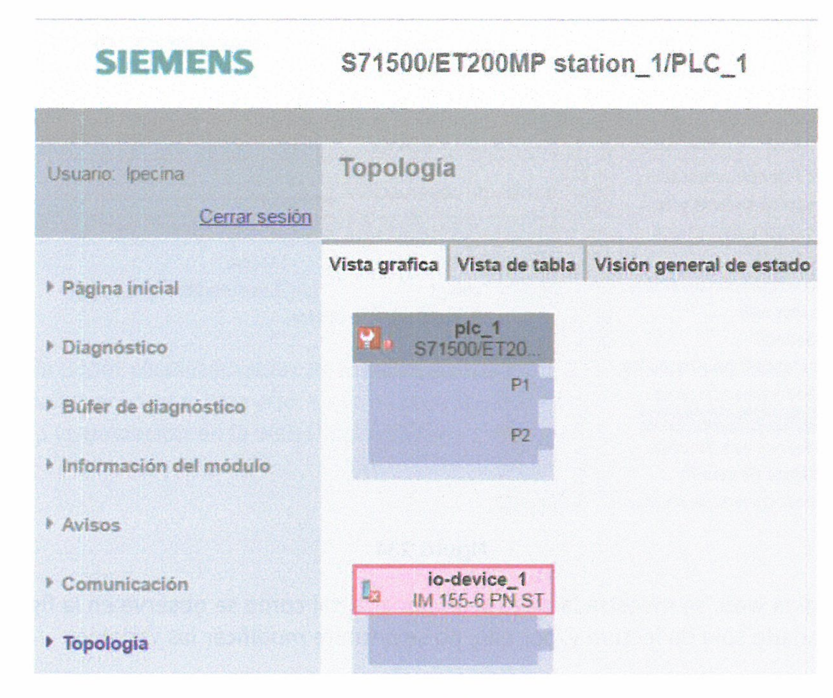


Figura 232

Estado de variables

Desde aquí se visualizan el estado de las variables del PLC y también se modifican sus valores. En la figura 233 se muestra alguna variable de las utilizadas en el programa.

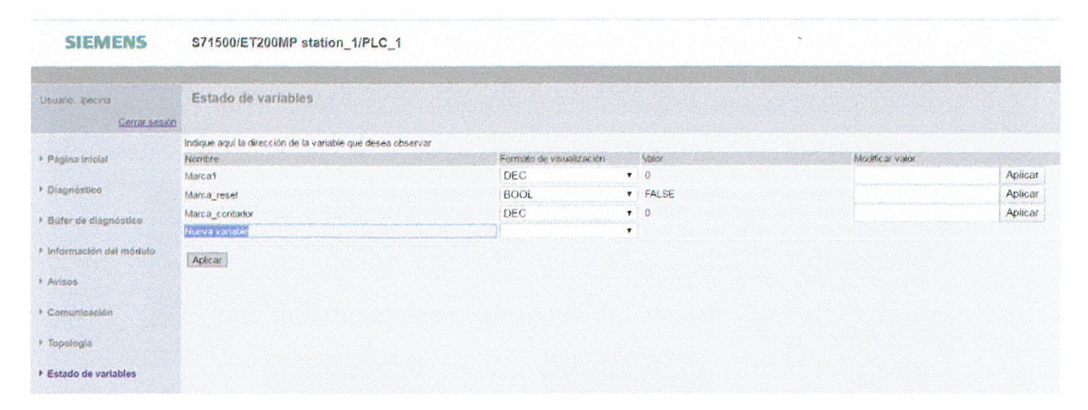


Figura 233

Se recomienda utilizar la opción de *Tablas de observación*, que se estudia a continuación.

Tablas de observación

En este apartado se visualizan aquellas tablas de observación autorizadas para publicarse en web en TIA PORTAL. Para realizar dicha configuración, hay que ir a las propiedades del PLC dentro del apartado *Servidor web/Tablas de observación*. Allí se selecciona la tabla que se desee publicar y se indica si puede solo leerse o también se va a permitir escribir (figura 234). No se visualiza en la web al menos que se haya elegido una tabla para observar.

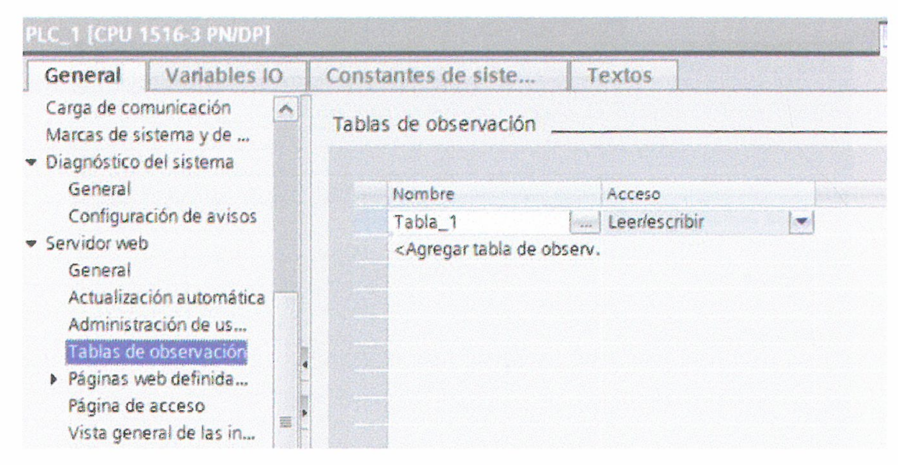


Figura 234

Después, en la página web, se muestra la tabla seleccionada, tal como se observa en la figura 235. En este caso se ha seleccionado solo de lectura y, por ello, no se permite modificar las variables.

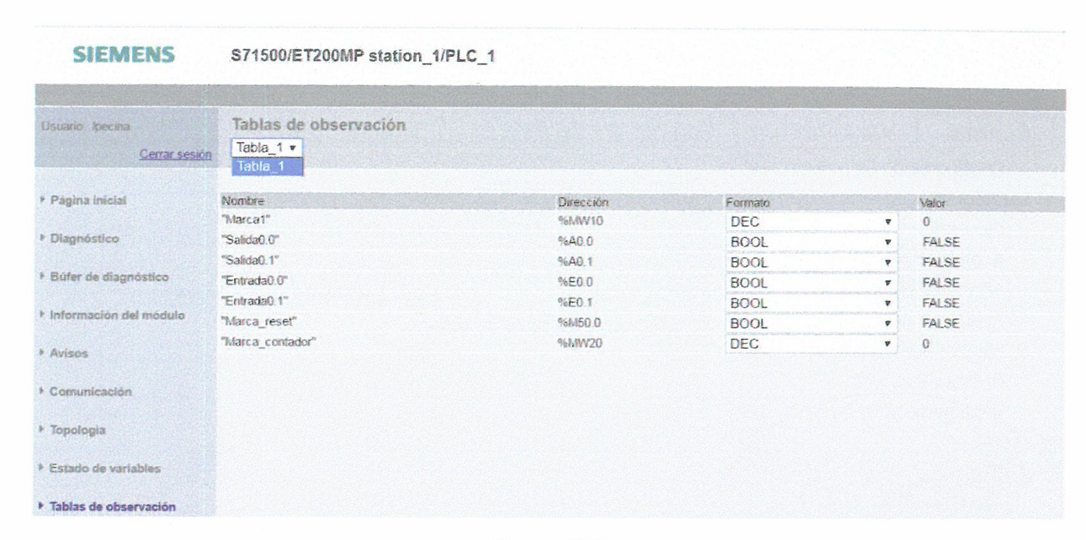


Figura 235

En la figura 236 se aprecia que sí se pueden **modificar las variables**, al haber sido configurada en modo lectura/escritura.

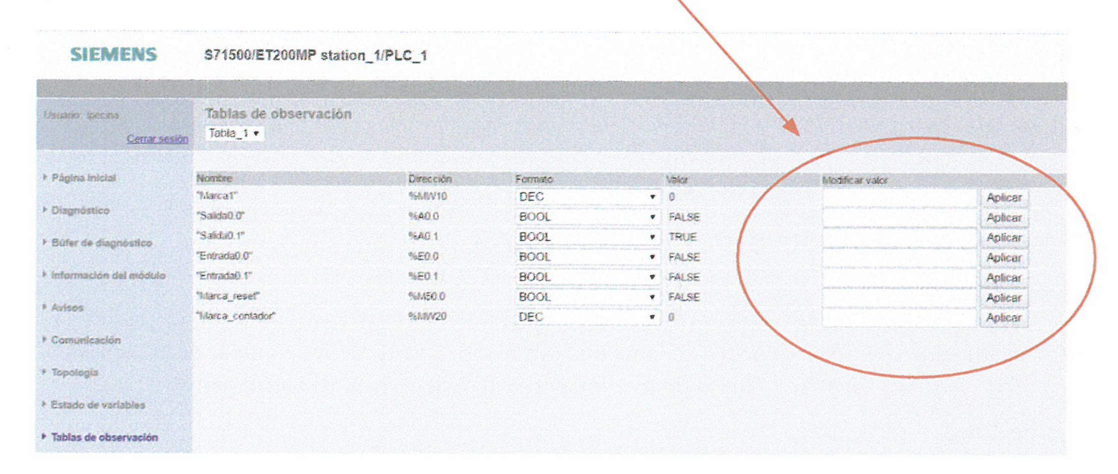


Figura 236

También es posible poder visualizar tablas de forzado en el *display* y modificar las variables, si se ha configurado en modo lectura/escritura. Para ello, se va a las propiedades del PLC, a la opción *Display,* y se hace lo mismo que para la visualización en la web (figura 237).

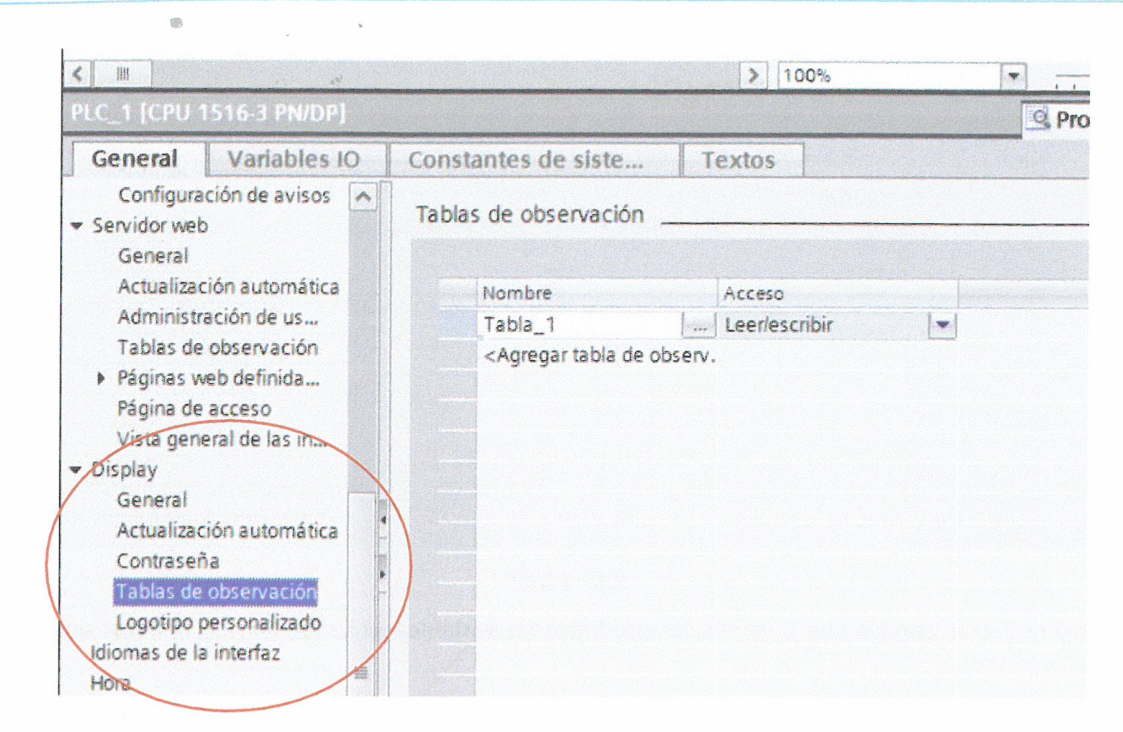


Figura 237

Páginas de usuario

Desde aquí se puede acceder a una página web que previamente se habrá creado. Estas páginas web se crean para interactuar con el PLC desde Internet.

Se hablará extensamente de este apartado en otro momento.

Navegador de archivos

Desde el explorador de archivos, se guardan datos directamente en la memoria de la tarjeta de la CPU o se cargan desde ella. Lo que se muestra en la página inicial del navegador de archivos es el contenido de la tarjeta del PLC. Algunos de esos archivos no pueden leerse ni borrarse, afortunadamente.

Registro de datos

En esta página se puede conseguir que se muestren todos los registros (DataLog) creados en el proyecto.

Unidad 6 Ejercicios prácticos de HMI

En esta unidad veremos:

17. Pantallas gráficas (HMI). Configuración de la pantalla
18. Intercambio de datos entre HMI y PLC
19. Descripción de las funciones disponibles en WinCC
20. HMI Avisos
21. HMI Recetas
22. Fábrica de bebidas

17. Pantallas gráficas (HMI). Configuración de la pantalla

Enunciado

Se va a realizar la conexión y configuración de una pantalla SIMATIC HMI KTP 700 BASIC conectada a un PLC S-7 1516 3PN/DP.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una pantalla SIMATIC HMI KTP 700 BASIC 6AV2 con referencia 123-2GB03-0AX0.
3. *Software* WinCC V13 Advanced.
4. Un cable de Profinet para las conexiones.

Realización

Se debe partir de una configuración de un PLC 1516 en TIA PORTAL. Una vez configurado el PLC, se debe añadir la pantalla gráfica. Esta debe poseer una dirección IP ubicada dentro del mismo grupo que el autómata y también deberá contar con un nombre dentro de la red de Profinet. La dirección y el nombre se pueden pasar de dos formas diferentes: utilizando *Accesos online* desde TIA PORTAL o desde la propia pantalla. El método mediante el acceso *online* ya se ha utilizado otras veces. Se encuentra en el árbol del proyecto y se debe buscar la interfaz que se utiliza para comunicar la programadora/PC. Se actualizan los dispositivos accesibles y, cuando encuentra la pantalla de HMI, se accede a *Online y diagnóstico*. Luego, en el área de la derecha y desde *Funciones,* se le asigna la dirección IP deseada y un nombre, si se quiere cambiar. Este proceso se puede observar en la figura 238. Hay que unir el PLC, la pantalla y el PC con el cable de Profinet.

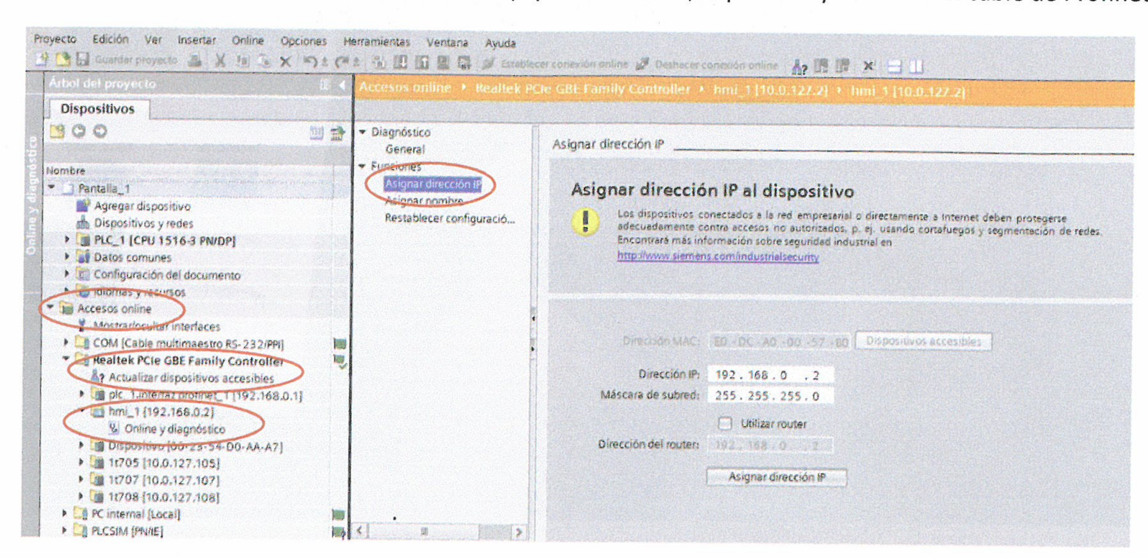


Figura 238

Para cambiar la dirección desde la propia pantalla, se entra en ella y se accede a las diversas ventanas para cargar la dirección y el nombre.

Al conectar, aparecerá la ventana mostrada en la figura 239. Pulse en *Settings* para entrar en las opciones de configuración.



Figura 239

Seguidamente, dentro Network Interfaz, se introduce la dirección y la máscara de subred. También se puede cambiar el nombre en esa misma pantalla más abajo (figura 240).

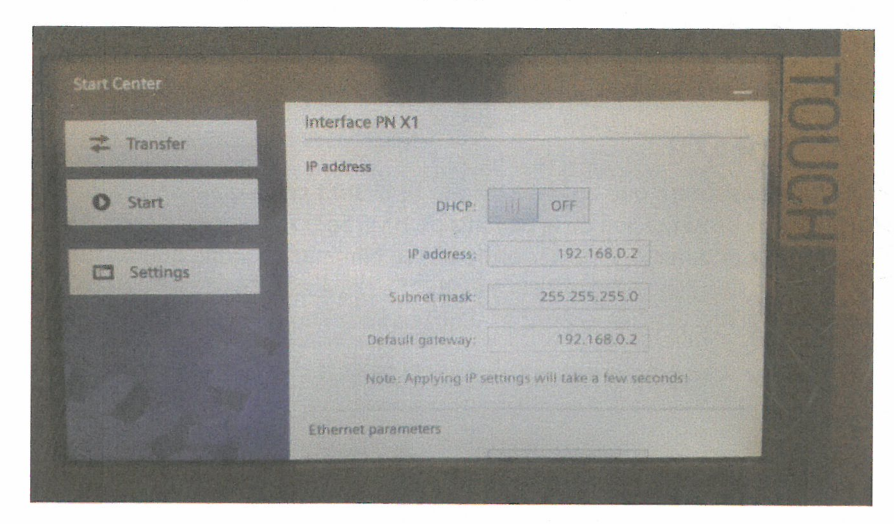


Figura 240

Desde el menú *Transfer Settings,* se puede poner el modo de transferencia automática y, de este modo, se transferirá siempre un fichero de WinCC a la pantalla. Esa ventana se mostrará en la figura 241.

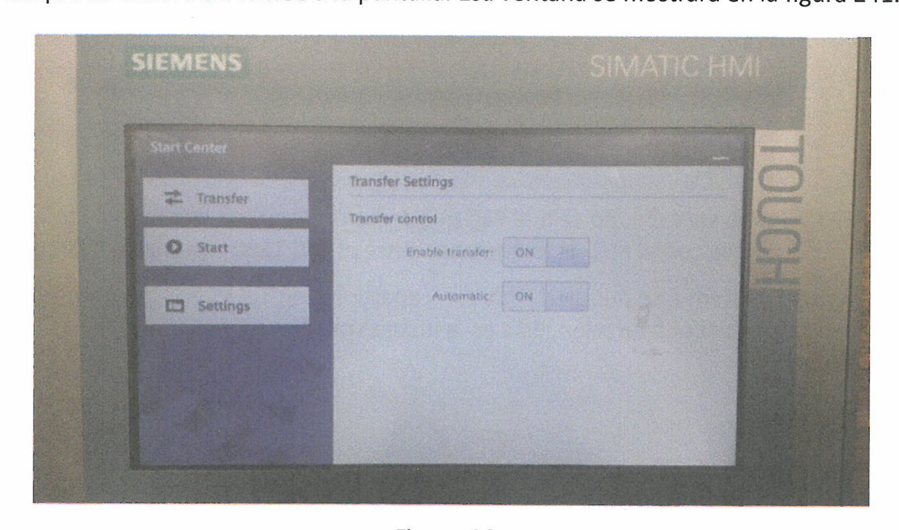


Figura 241

Configuración de la pantalla de HMI en TIA PORTAL

Lo que se acaba de realizar se puede hacer posteriormente, durante el proceso de configuración en TIA PORTAL. Si se hubiera hecho antes, se escribe en el proyecto de TIA PORTAL la dirección y el nombre que se le ha puesto en ese momento. Se ha de pensar que el PLC deberá tener una dirección del mismo grupo que la pantalla.

Para comenzar la configuración en TIA PORTAL, se debe ***Agregar dispositivo*** nuevo y buscar la pantalla en el menú HMI, como se aprecia en las figuras 242 y 243.

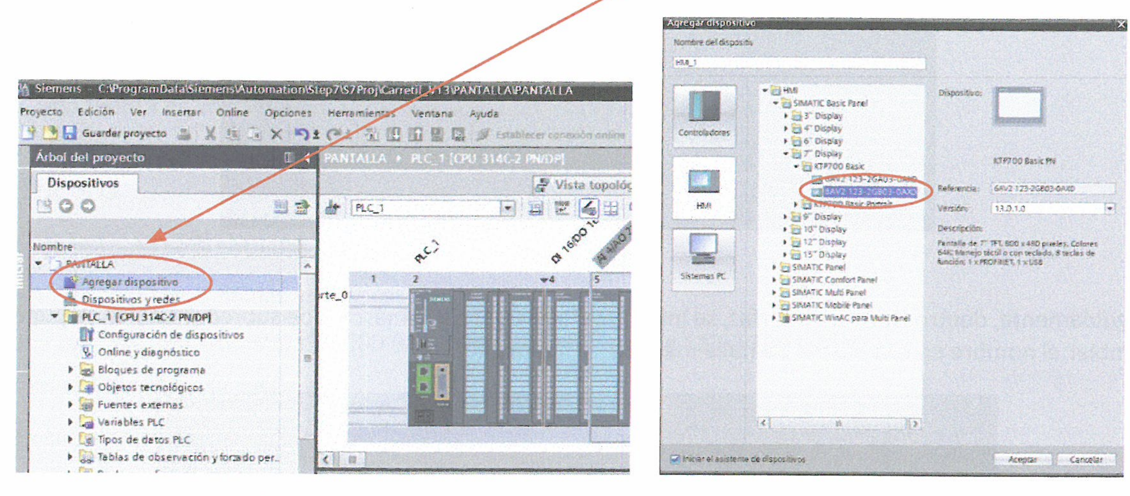


Figura 242 **Figura 243**

Ahora se puede seguir el asistente que aparece. Según se avanza con *Siguiente,* aparecerán una serie de ventanas para configurar la presentación de la pantalla de HMI. Solo será imprescindible la primera, aunque se pueden aceptar todas por defecto y, posteriormente, se eliminará aquello que no interese. La primera ventana se refiere a la conexión entre la pantalla y el PLC. Se debe examinar y seleccionar con quién estará conectado. Se selecciona el PLC y aparecerá la conexión, como se indica en las figuras 244 y 245.

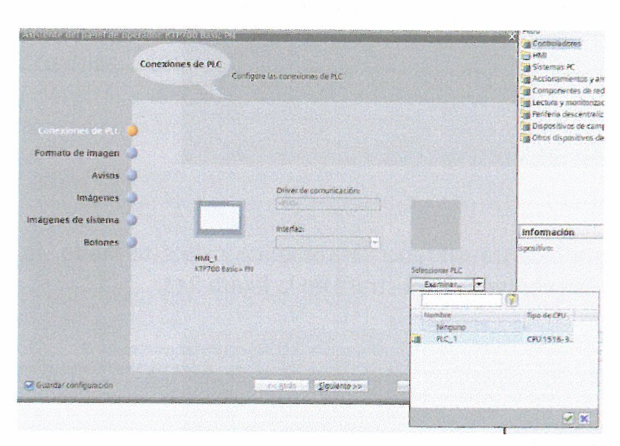


Figura 244

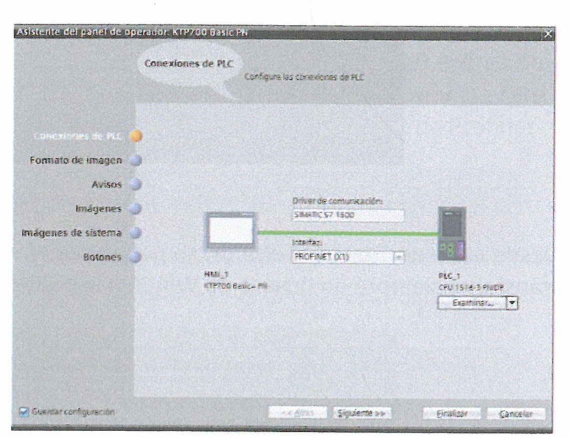


Figura 245

Si, al examinar en la primera ventana, no sale el PLC que se ha configurado en el proyecto, significa que la pantalla no resulta compatible con el PLC. Es lo que pasa entre **el PLC 1500 y la pantalla 177B**, por ejemplo.

En la última ventana del asistente, se pulsa en *Finalizar* y aparecerá en TIA PORTAL la imagen principal de la pantalla para proseguir con su configuración. Esta es la imagen raíz (figura 246).

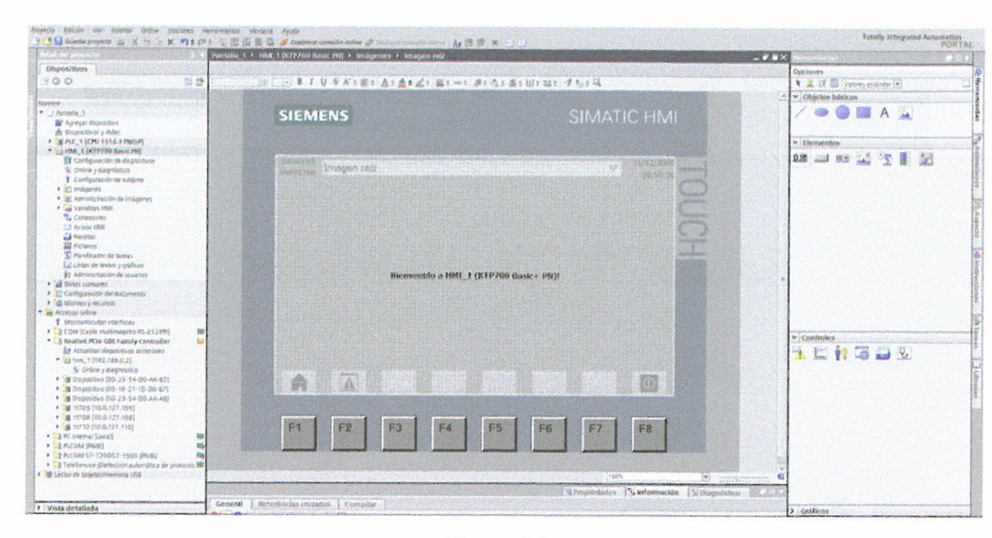


Figura 246

Ahora, desde *Dispositivos y redes,* se cambiará la dirección de la pantalla de acuerdo con lo hecho anteriormente, así como su máscara de subred y el nombre de dispositivo de Profinet (figura 247).

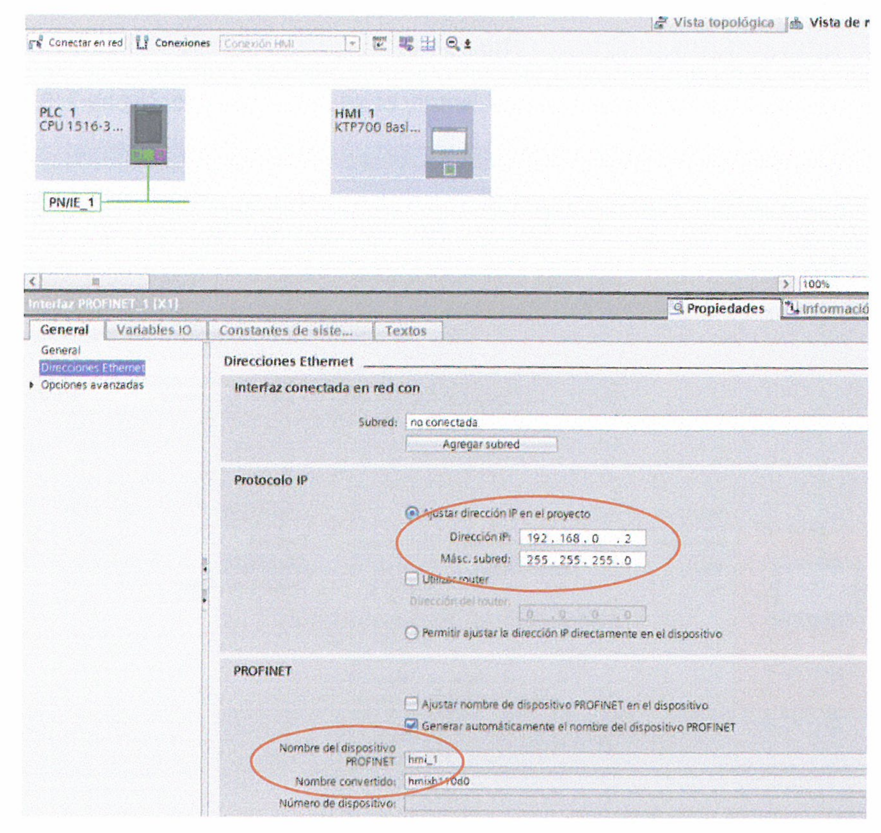


Figura 247

Se establece la red entre el PLC y la pantalla y se envía hacia la pantalla para comprobar que todo funciona bien. Conviene antes **compilar** para detectar los posibles errores que pudieran producirse. Para enviar, se toca en la pantalla y se activa el icono de **enviar** (figura 248).

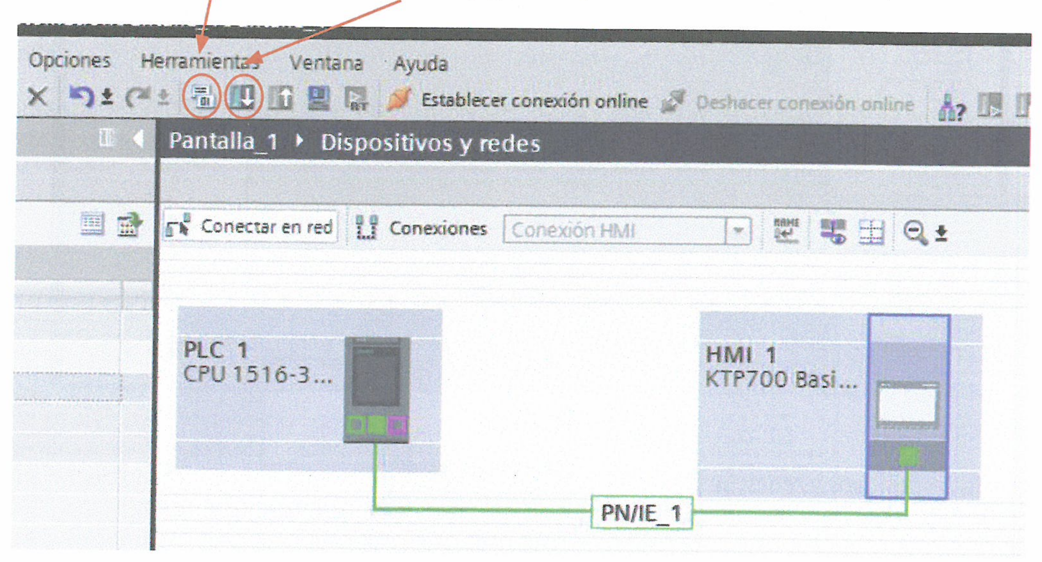


Figura 248

Se *Inicia búsqueda* y encontrará la pantalla de HMI, entonces, se pulsa en *Cargar* (figura 249).

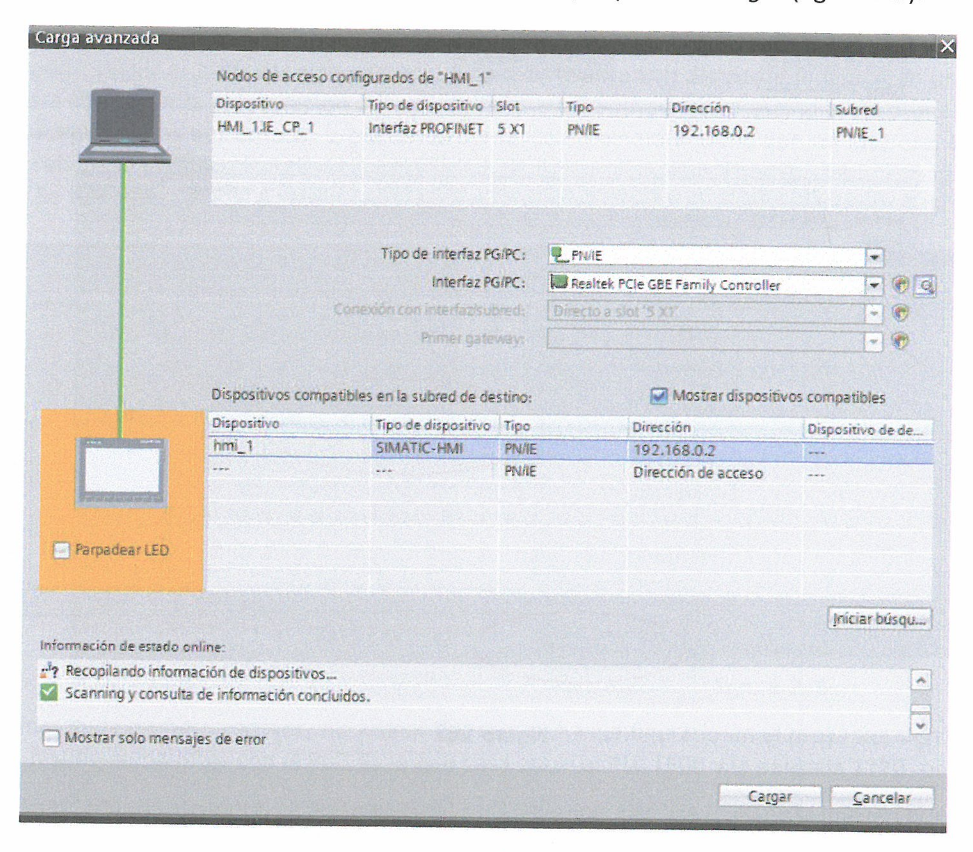


Figura 249

La imagen que aparece en la pantalla se muestra en la figura 250.

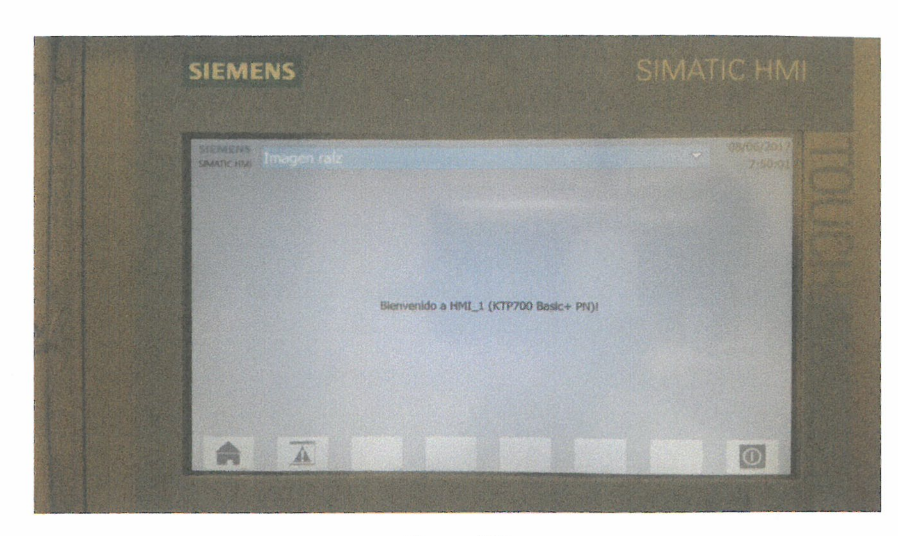


Figura 250

Se pueden eliminar algunas de las cosas que surgen en la pantalla; simplemente hay que seleccionarlas de una en una y pulsar *Suprimir* desde TIA PORTAL. En la pantalla de HMI, si se pulsa el botón , la pantalla se desplazará al menú principal desde el que se podrá volver a entrar *(Start)*, ir a la configuración de la pantalla *(Settings)* o esperar una transferencia desde el PC *(Transfer)*.

A continuación, se accede desde la pantalla al PLC y viceversa, es decir, el trabajo de usar el WinCC. Al abrir la pantalla, habrá salido en el árbol del proyecto y las diversas opciones para trabajar con ella. En la figura 251 se observan el árbol de carpetas y las opciones de la pantalla.

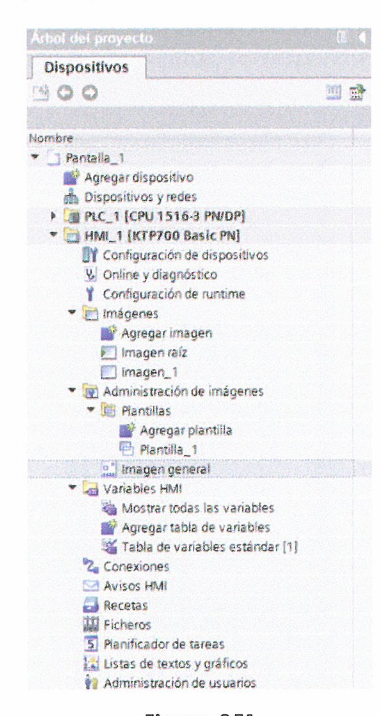


Figura 251

Desde el menú *Imágenes,* se pueden agregar más imágenes. La que sale por defecto es la imagen raíz. Desde *Administración de imágenes,* se puede crear una imagen plantilla y, en *Imagen general,* se visualizan todas las pantallas de avisos, que se estudiarán más adelante. Todo lo que se inserte en la imagen plantilla aparecerá en todas las imágenes. Para que esto sea así, en la imagen creada, habrá que decirle qué plantilla se desea obtener, aunque solo exista una plantilla creada; de lo contrario, no se obtendrá ninguna plantilla. La imagen raíz es la primera que saldrá en la pantalla. Para borrar elementos no deseados ubicados en la plantilla, solo se puede hacer desde la imagen plantilla.

En la figura 252 aparece la nueva situación de la pantalla después de haber borrado lo que no era de interés para nuestro proyecto y que se había colocado a través del asistente. Se ha añadido un nuevo logotipo con la opción de copiar y pegar.

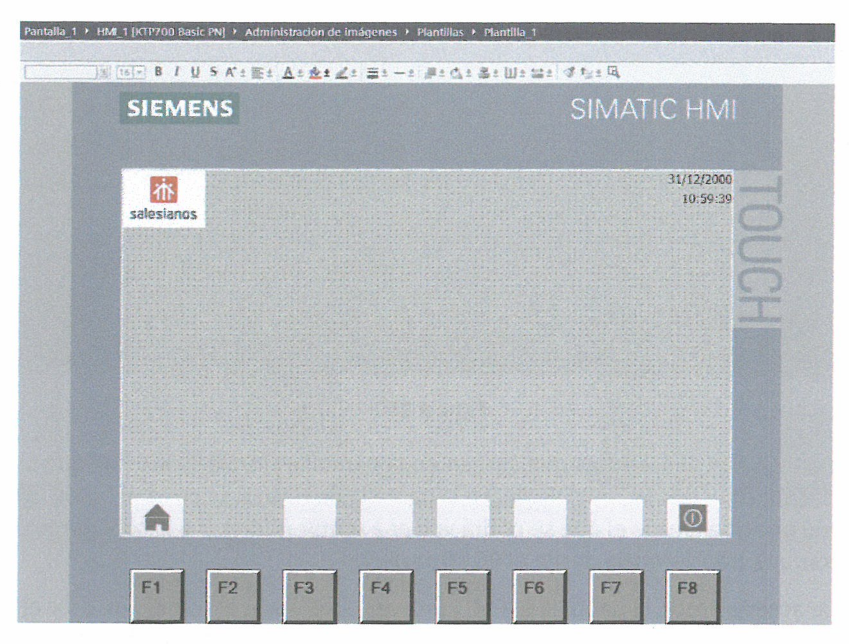


Figura 252

En los próximos ejercicios se desarrollarán las diferentes posibilidades que se pueden obtener con WinCC y las pantallas de HMI.

18. Intercambio de datos entre HMI y PLC

Enunciado

Una vez que se tiene configurada la pantalla con el PLC, ha llegado el momento de interactuar entre la pantalla de HMI y el PLC y su periferia, si la tuviera. Para ello, se creará un botón en la imagen raíz de la pantalla de forma que active una salida del PLC. Desde el PLC, se deberá activar un indicador en la pantalla cuando se active una salida del PLC.

Material necesario

1. Una CPU 1516-3PN/DP.
2. Una pantalla SIMATIC HMI KTP 700 BASIC 6AV2 con referencia 123-2GB03-0AX0.
3. *Software* WinCC V13 Advanced.
4. Un cable de Profinet para las conexiones.

Realización

Una vez configurada la pantalla y en conexión con el PLC, se puede empezar a utilizarla para intercambiar datos con el PLC.

Las variables empleadas en el PLC se podrán usar en la pantalla. Las variables del PLC aparecen en *Variables del PLC* y las de la pantalla, en *Variables HMI*. Fácilmente se podrán tomar las variables del PLC para usarlas en la pantalla.

En la parte derecha se dispone del menú *Herramientas*, correspondientes a los elementos de dibujo que se emplean en las imágenes (figura 253).

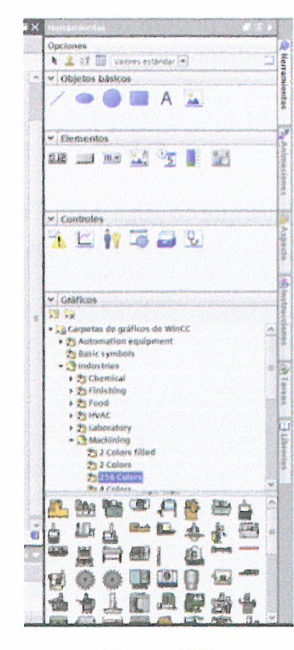


Figura 253

Todos los objetos se pueden llevar a la pantalla arrastrando con el ratón. Para este primer ejemplo, se va a colocar en la pantalla un *Botón* de la carpeta *Elementos*. Al colocarlo y tocar sobre él, se abren en la parte inferior sus propiedades (figura 254).

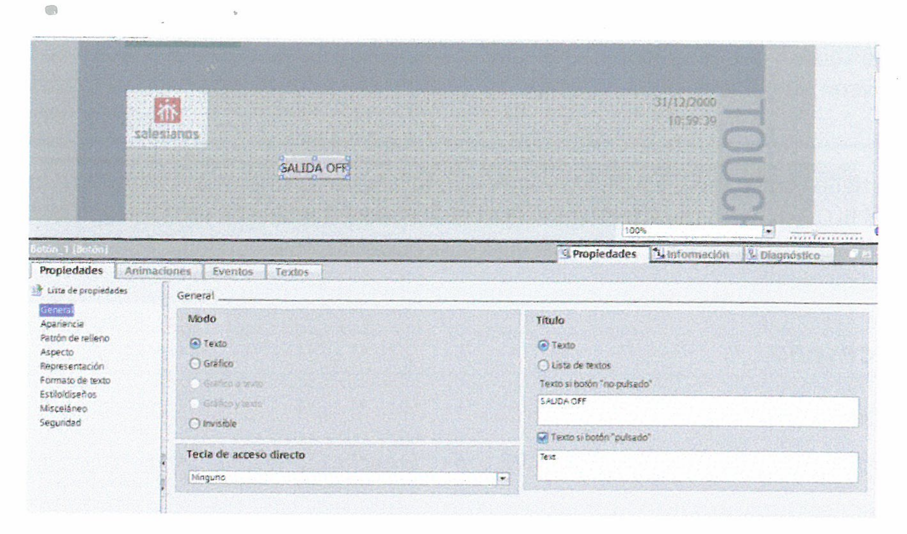


Figura 254

Se muestran las siguientes carpetas: *Propiedades, Animaciones, Eventos* y *Textos.* Cada una hace referencias a distintas consideraciones sobre el elemento *Botón*. Resulta muy importante **pulsar sobre el elemento *Botón*** para que salgan las propiedades de dicho elemento y no de la imagen.

En las *Propiedades* se pueden cambiar tamaños, colores y otros aspectos propios de la presentación externa en pantalla. En *Animaciones,* se presenta la posibilidad de incorporar movimientos y su forma de visualización. En *Eventos* se pone la acción que se desea adquiera el botón en caso de tener que ser accionado (como control). En *Textos* se hallan las acciones sobre los textos.

Elementos como indicadores

En este primer ejemplo se va a utilizar el botón como indicador, no como control. Se asociará a ese botón una salida del PLC y, cuando se encuentre a 1, se tornará en verde y, a 0, rojo. Es necesario crear en el PLC la tabla de variables que se empleará. Luego se podrán tomar esas variables para la pantalla. En el PLC se escribe un programa que active esa salida mediante una entrada del PLC:

```
OB1 PLC: U Entrada0.0   // E0.0
         = Salida1.0    // A1.0
```

En propiedades del *Botón,* se borra el texto para que no salga texto y, en *animaciones/visualización/apariencia,* se asocia la variable deseada (A1.0) y se establece que, cuando sea 0, el color del botón pase a rojo y cuando sea 1, a verde (figuras 255 y 256). Para seleccionar la variable deseada, al accionar sobre el icono [...], aparecerá el árbol de las variables, tanto del PLC como de HMI. Se selecciona la variable del PLC. Tras hacerlo para usarla en la pantalla, quedará también integrada como variable de HMI (figura 257).

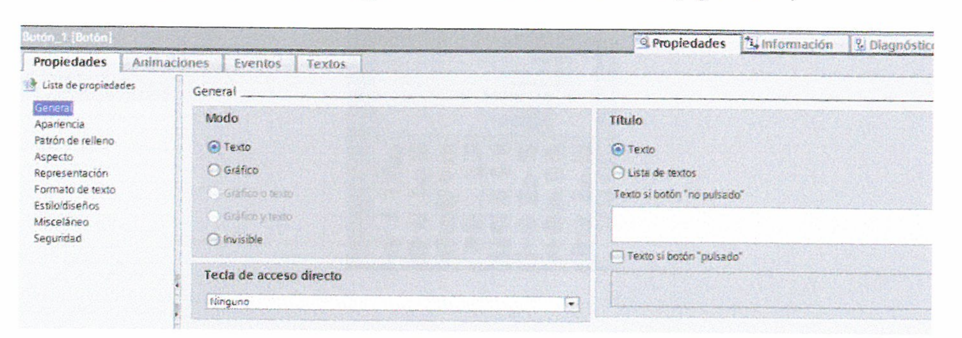


Figura 255

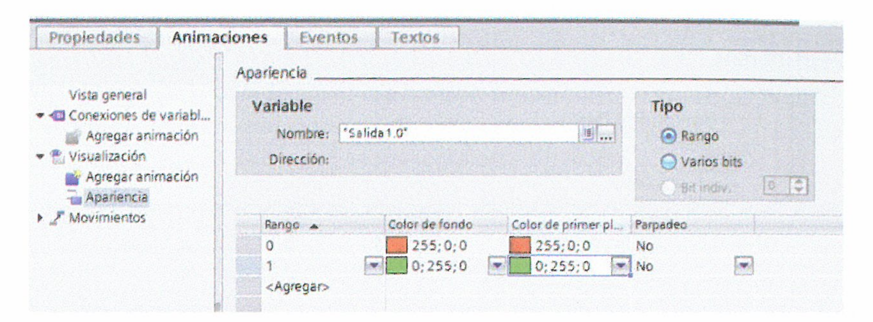
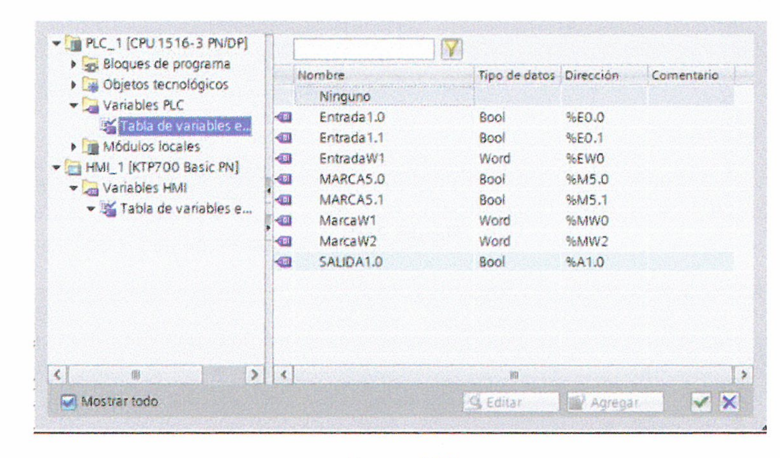


Figura 256



Nombre	Tipo de datos	Dirección	Comentario
Ninguno			
Entrada1.0	Bool	%E0.0	
Entrada1.1	Bool	%E0.1	
EntradaW1	Word	%EW0	
MARCA5.0	Bool	%M5.0	
MARCA5.1	Bool	%M5.1	
MarcaW1	Word	%MW0	
MarcaW2	Word	%MW2	
SALIDA1.0	Bool	%A1.0	

Figura 257

A continuación se carga sobre la pantalla y se observará que, al pulsar la entrada del PLC (E0.0) que activa la salida (A1.0), el botón se pondrá verde (la salida será 1) y, al soltar, se cambiará a rojo (la salida será 0).

Elementos como controles

Ahora se va a colocar otro botón, en este caso como control. Se asociará a una salida del PLC, de forma que, al pulsar, se active y, al soltar, se desactive dicha salida. Se pondrá un texto (OFF) cuando no se pulse y otro (ON) cuando se active.

Para ello se colocará otro botón sobre la imagen raíz y se cambia el texto como se indica en la figura 258.

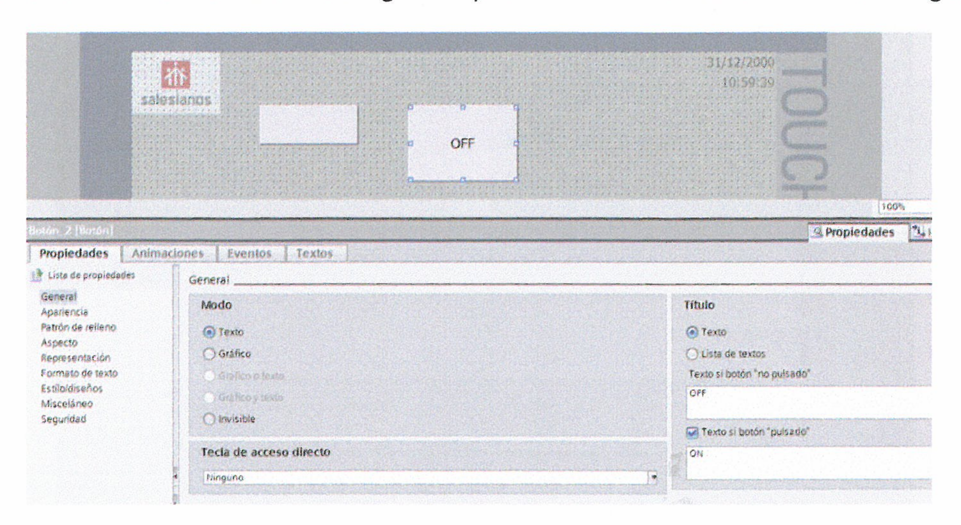


Figura 258

Desde la carpeta *Eventos,* se selecciona la opción *Pulsar*, se busca la función que se desea hacer al pulsar y se asocia una variable a esa acción de pulsar. Este procedimiento se detalla en las figuras 259 y 260.

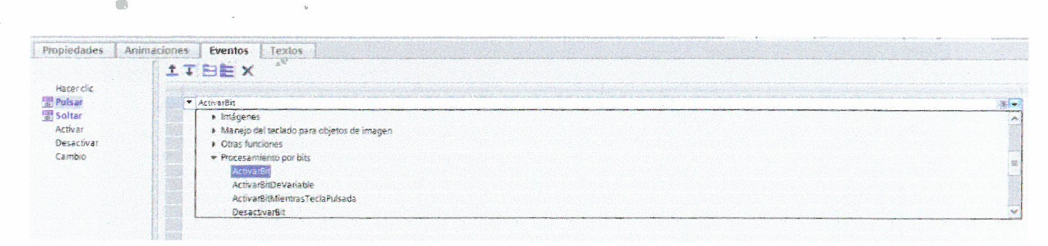


Figura 259

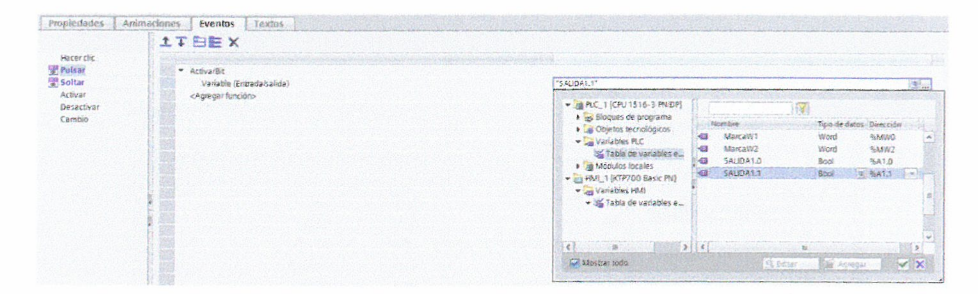


Figura 260

Lo mismo se deberá hacer con la acción de *Soltar* (figura 261).

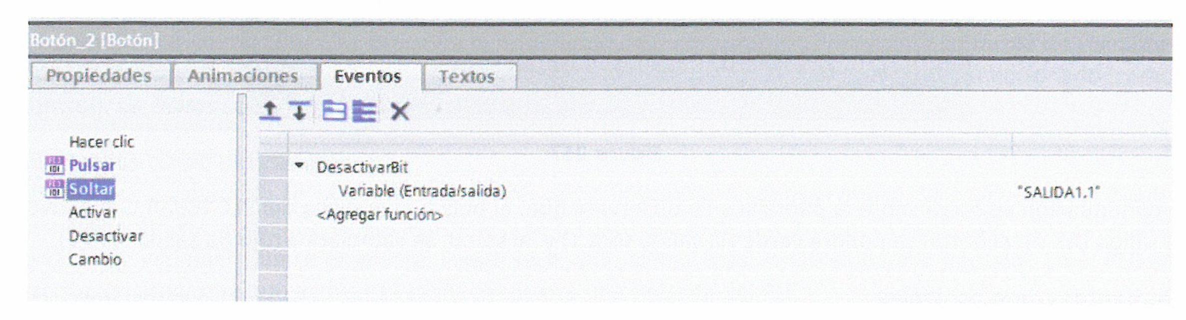


Figura 261

En la figura 262 se puede ver el aspecto de la pantalla una vez configurados los dos botones.

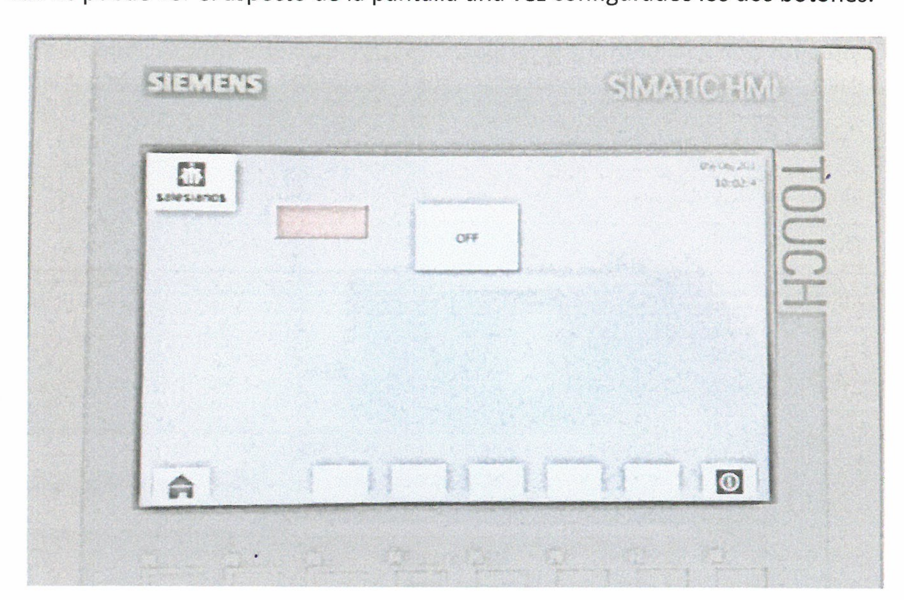


Figura 262

El botón creado en el ejemplo anterior actúa como un pulsador. Se van a probar las diferentes acciones que existen sobre el botón. Ya se ha probado la acción *Pulsar* y *Soltar*. La acción *Hace clic* activa la variable o la acción deseada al pulsar sobre el botón. Se va a utilizar esta acción para cambiar de una imagen a otra y, para

ello, se deberá añadir otra imagen más. Ahora se tiene la *Imagen raíz* y la *Imagen_1*. Se añade un botón en la imagen raíz y otro en la imagen_1. A los dos botones se les va a asignar la función de *ActivarImagen*; en el caso del botón de la imagen raíz se le asociará esa función señalando que active la imagen_1 para que avance a esa imagen. Al botón de la imagen_1 se le indicará que vaya a la imagen raíz. En cada botón se muestra un texto explicativo. En las figuras 263 y 264 se presenta esta situación.

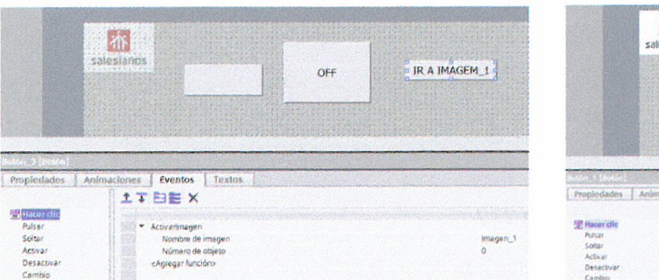


Figura 263

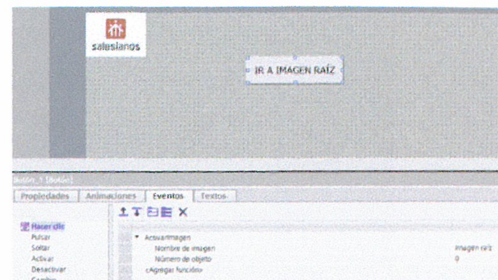


Figura 264

La acción *Cambiar* es como ejecutar un set en la variable deseada. Como ejemplo se van a colocar dos botones en la imagen raíz. Uno activará un bit y el otro lo desactivará. El efecto será como un set/reset de esa variable. En las figuras 265 y 266 se resalta el resultado.

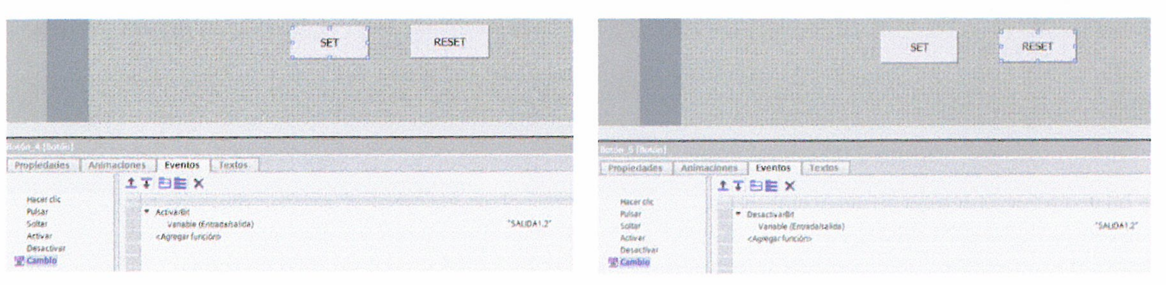


Figura 265 Figura 266

Otro elemento para utilizar es el interruptor. A este se le va a asociar la acción de *Conmutar ON* y *Conmutar OFF*. También se pondrá rojo cuando se encuentre en OFF (0) y verde cuando lo esté en ON (1). En las figuras 267-269 se representa dicho procedimiento.

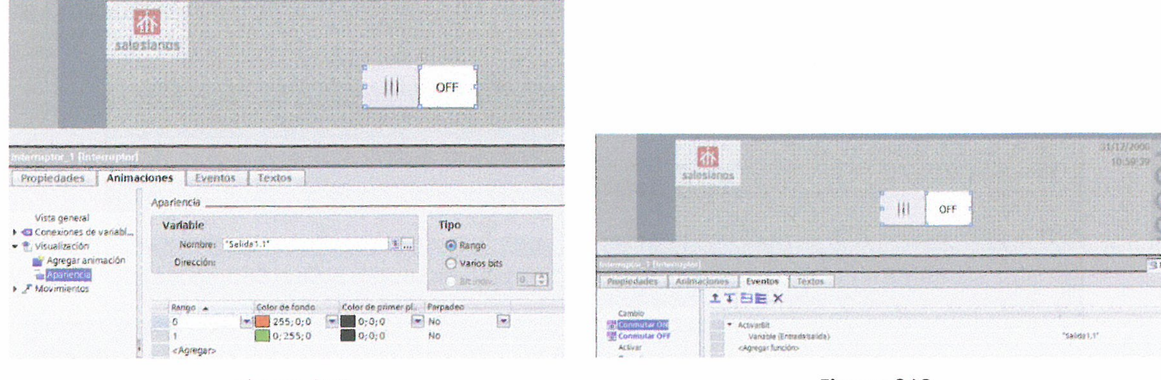


Figura 267 Figura 268

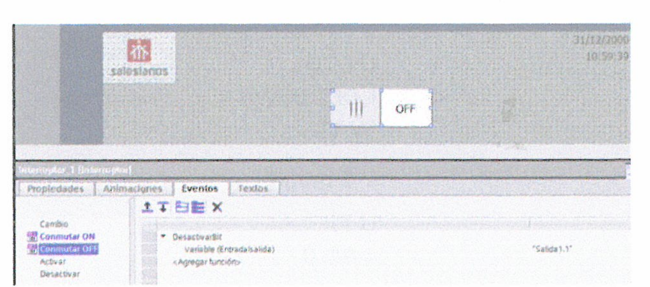


Figura 269

Si se le asigna la acción de *Activar* y *Desactivar,* se deberán añadir las propiedades del proceso, tal como se aprecia en la figura 270.

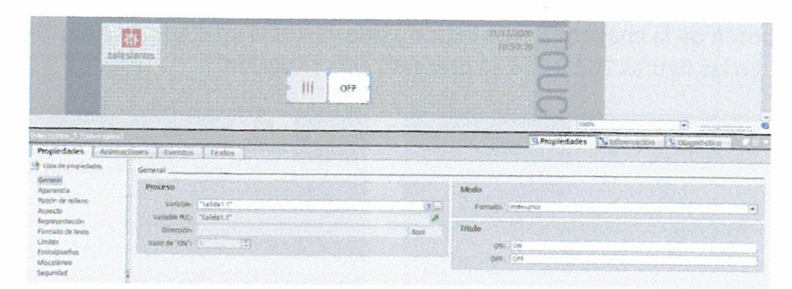


Figura 270

En este caso habrá que asociar las acciones deseadas (figuras 271 y 272).

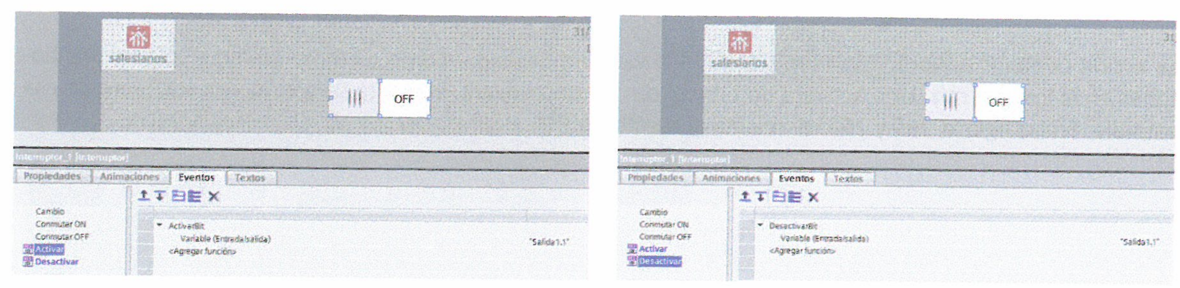


Figura 271 **Figura 272**

Se ha visto que, además de las acciones sobre los botones y otros elementos, existen diversas funciones que pueden ejecutar esos mismos elementos. Las funciones resultan muchas y variadas y se debería realizar un repaso de todas ellas. En este texto se expondrán algunas de esas funciones, pero no todas.

Campo de entrada y salida

Otro elemento que se puede utilizar es el campo de entrada y salida, en el que se escriben valores o se leen valores de variables. En el ejemplo descrito se utilizará para cambiar la variable de tiempo de un temporizador. Se debe seguir el siguiente procedimiento: crear un DB en el autómata y añadir un dato (denominado *tiempo*) con constante de tiempo S5 y un tiempo de inicio de un segundo (S5TIME----- S5T#1s), como se observa en la figura 273.

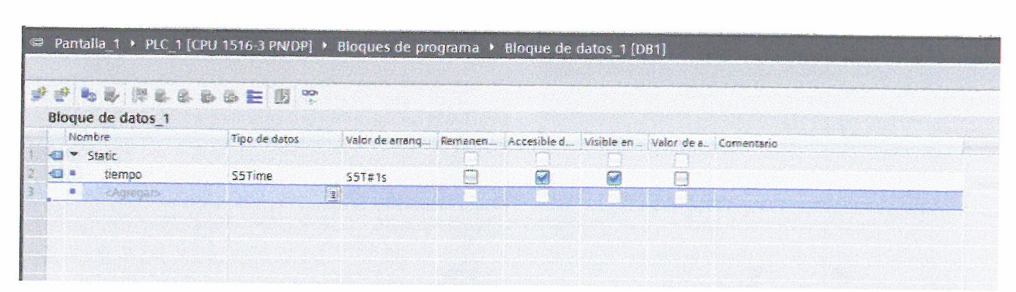


Figura 273

Se escribe el OB1:

```
U E0.0
L DB1.DBW0
SV T1

U T1
= A4. 0
```


En WinCC (imagen raíz). Se coloca un campo-ES (0.12).

Se toca dos veces sobre él y se accede a la carpeta *General* en *Propiedades*. En *Proceso,* se busca la variable del DB. Para ello se debe acceder al DB1 desde la ventana que aparece al seleccionar la variable. Véase la figura 274.

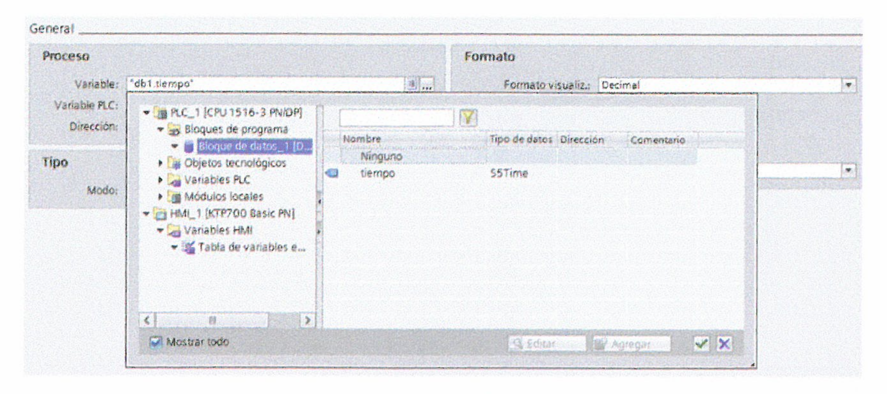


Figura 274

Seguidamente, se debe colocar el modo del tipo de dato, el formato de visualización y el de representación, tal como se indica en la figura 275.

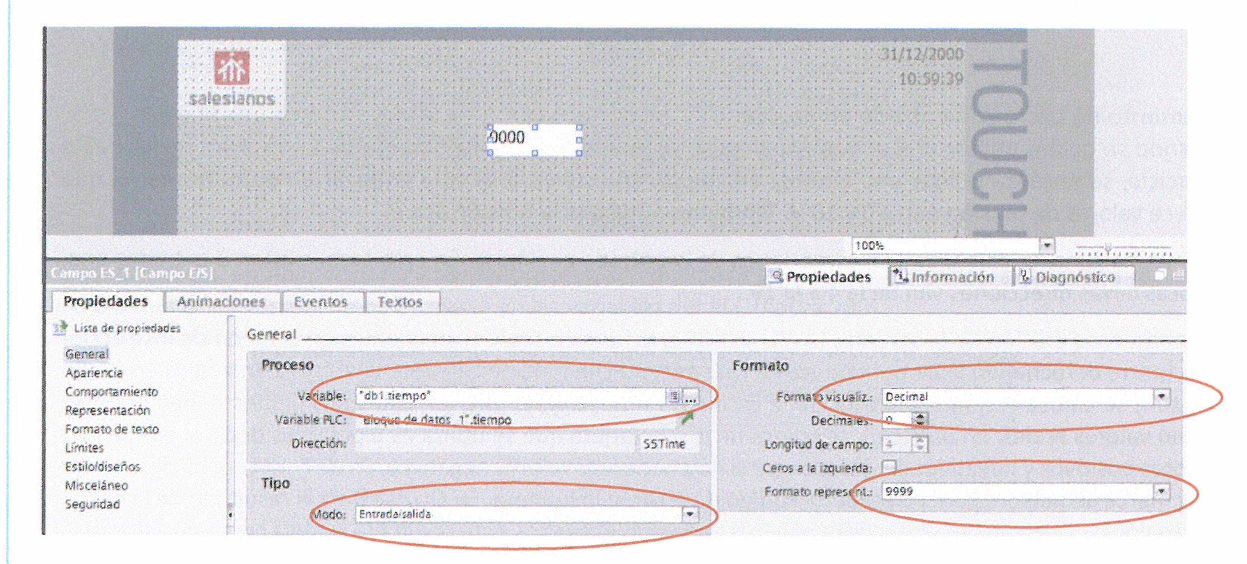


Figura 275

En *Eventos*, la acción será *Activar* y la función, *ActualizarVariable*. En la ID de actualización, se debe buscar la variable tiempo del DB1 que se desea modificar. En la figura 276 se refleja esta situación.

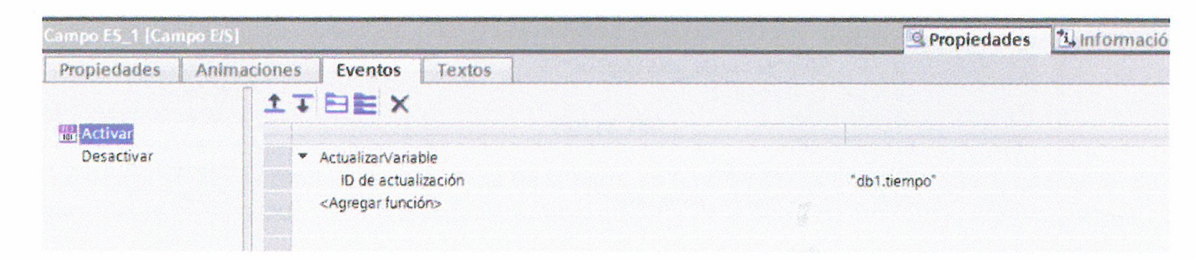


Figura 276

Desde la pantalla, se introduce un valor de tiempo en milisegundos y, al activar *Enter,* se carga el nuevo valor en el DB. Para poder ver el DB y su nuevo valor, se activan las gafas, de modo que se observará cómo cambia dicho valor. Al arrancar el temporizador con la entrada E0.0, tomará el nuevo valor de tiempo. En la figura 277 se muestran la pantalla y el DB con las gafas.

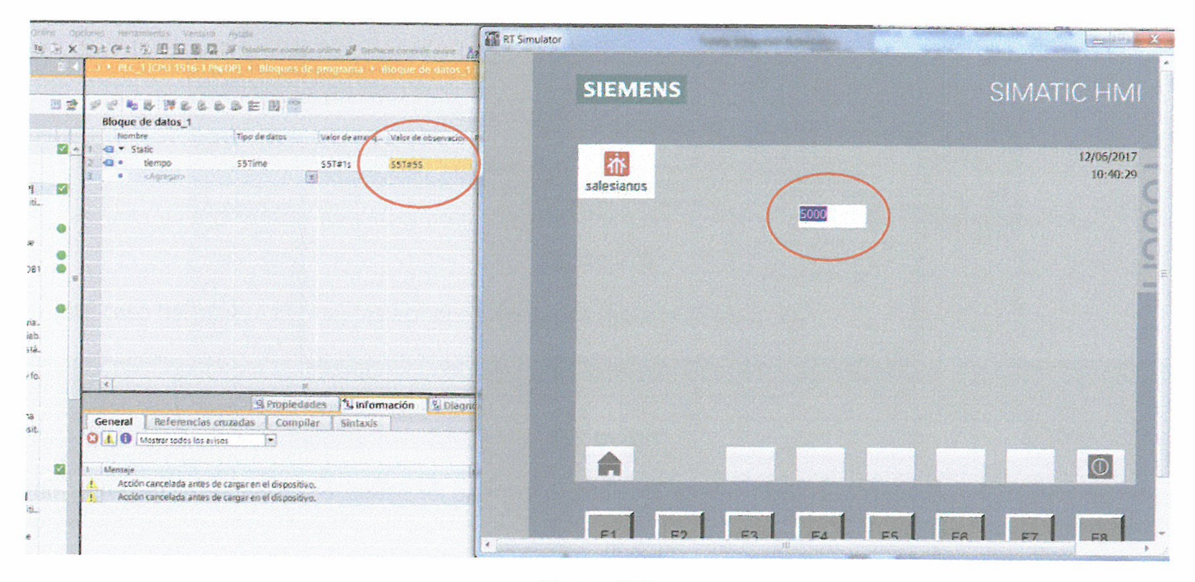


Figura 277

Elemento de barra con entrada analógica

Cuando se quiere utilizar una entrada analógica, se puede utilizar el elemento de barra. Para realizar este ejercicio, se necesita utilizar una entrada analógica. En este caso se va a emplear un potenciómetro, que ofrece valores de tensión entre 0 y 10 V. También se utilizará la función *Scale*.

Se han de conocer cuáles son las direcciones de la entrada analógica. En este caso, existen 8 entradas analógicas cuyas direcciones van de la 4 a la 19.

Estando en la edición de programas (AWL) del OB1, la función *Scale* se encuentra en el menú de instrucciones (parte derecha) en la carpeta *Conversión/Legacy*. Dicha función requiere la entrada analógica en *IN*. En *HI_LIM,* se coloca el límite superior de la escala y, en *LO_LIM,* el inferior. Estos dos límites deben ponerse como valores reales. En *BIPOLAR,* se coloca un 1 si la tarjeta que se utiliza es bipolar; es decir, si puede leer valores positivos y negativos. Se coloca un 0 si no puede leer valores negativos. En *RET_VAL,* se ha de insertar una marca de palabra para retornar un código de error, si lo hubiera. En *OUT* se deja el resultado de la entrada analógica ya escalado; se necesita una marca de doble palabra. Este resultado se deja como un valor real.

En el ejemplo se supone que la entrada analógica se corresponde con un tacómetro con el cual medir la velocidad de un motor cuya velocidad se sitúa entre 0 y 12.000 rpm.

En el OB1 del PLC se emplazan las siguientes órdenes:

```

      CALL  SCALE
        IN        :="Entrada_Analógica":P        %EW4:P
        HI_LIM    :=12000.0                      12000.0
        LO_LIM    :=0.0                          0.0
        BIPOLAR   :=0                            0
        RET_VAL   :="Error"                      %MW12
        OUT       :="Valor_escalado"             %MD20

```

No se debe olvidar poner una coma (punto, en este caso, como parte fraccionaria) y un 0, aunque no lleve decimales.

El valor que se va a representar en la pantalla será el resultado, es decir, la marca MD20.

Para ello, en la pantalla situamos el elemento de barra. Tocando sobre la barra, se accede a sus propiedades y, en *General,* se pone la escala deseada; en este caso, de 0 a 12.000. En *Escalas,* se marca *Escalar automáticamente* (figuras 278 y 279).

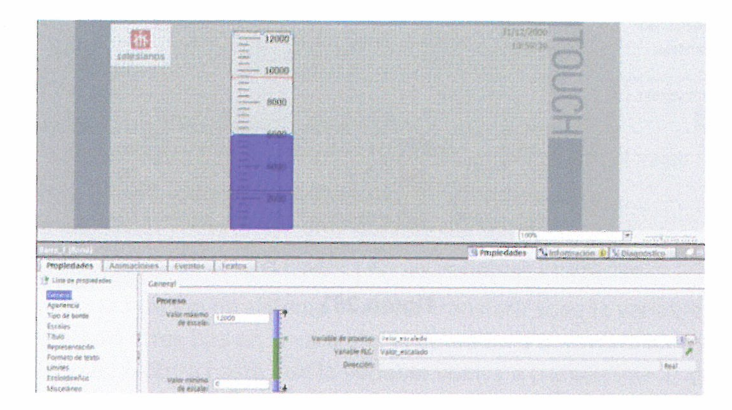


Figura 278

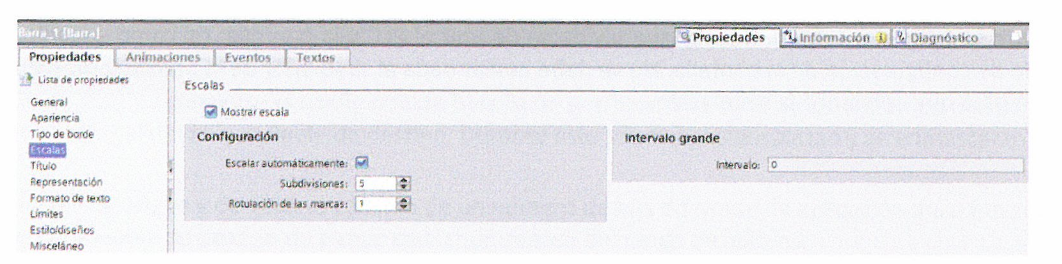


Figura 279

A continuación, en *Animaciones/Valor de proceso,* se conecta la variable MD20 *(Valor_escalado)* a la barra de la pantalla. La variable debe buscarse en las variables del PLC (figura 280).

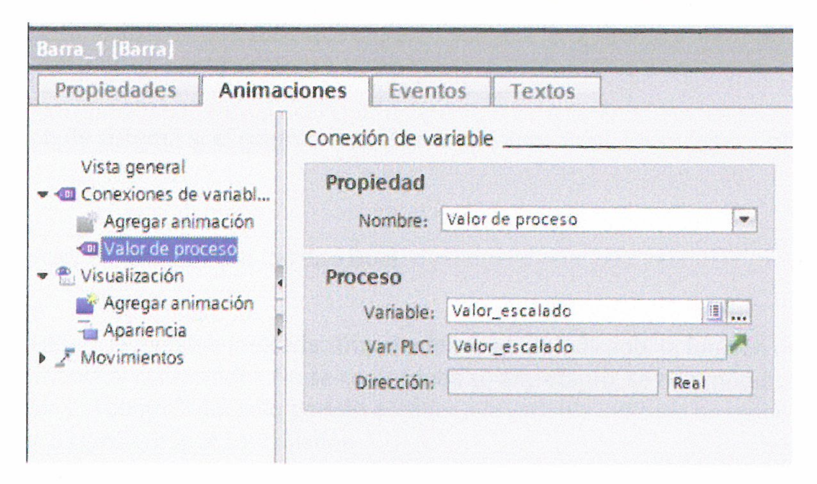


Figura 280

Si se desea, se puede cambiar la apariencia de la barra en función del valor de la entrada analógica. En la figura 281 se observa que se ha realizado un cambio de color de la barra en función de algunos tramos.

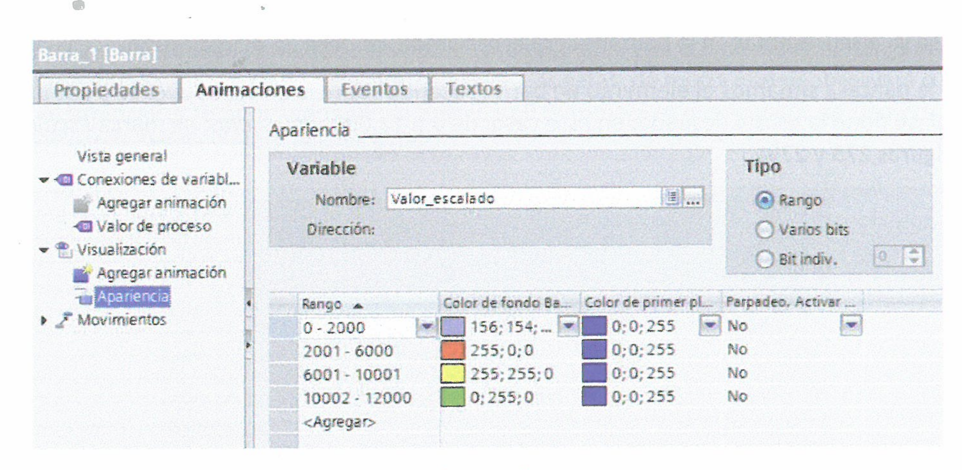


Figura 281

Simulador

Para activar el simulador de la pantalla, se debe activar el icono de *Simulación*, el mismo que se emplea para el simulador del PLC. Se puede dejar abierto el simulador del PLC o un PLC real. Para activar uno u otro, basta con situarse en el PLC o en la pantalla. Si se cuenta con un PLC real, el simulador de la pantalla actuará sobre el PLC que se tenga conectado. No hay que olvidarse de unir el PLC y la pantalla, tal como se indica en el apartado de configuración de la pantalla. **No se debe enviar** nada al simulador de la pantalla; simplemente, se abrirá para que comience la simulación.

19. Descripción de las funciones disponibles en WinCC

En la siguiente sección se describen las diferentes funciones disponibles para las acciones de los elementos u objetos.

Procesamiento de bits

InvertirBit

Invierte el valor de la variable indicada del tipo Bool. Si la variable cuenta con el valor 1 (TRUE), se ajusta a 0 (FALSE). Si la variable posee el valor 0 (FALSE), se ajusta a 1 (TRUE).

- ***DesactivarBit:*** pone a 0 (FALSE) el valor de una variable del tipo Bool.
- ***DesactivarBitEnVariable:*** sitúa un bit a 0 (FALSE) en las variables indicadas.

Tras modificar el bit indicado, la función de sistema vuelve a transferir toda la variable al controlador. No se comprueba si se han modificado otros bits en la variable. El operador y el controlador solo podrán acceder a la variable indicada en modo de lectura hasta que la variable vuelva a transferirse al controlador.

Nota

No utilice esta función de sistema si el controlador admite variables Bool. En su lugar, utilice la función de sistema *DesactivarBit*.

DesplazarYEnmascarar

La función de sistema convierte la configuración binaria de entrada de la variable de origen en una configuración binaria de salida de la variable de destino. Durante este proceso, se desplazan y se enmascaran bits.

Nota

Si las variables de origen y de destino constan de un número de bits diferente, la aplicación de la función de sistema en la variable de destino no puede causar un rebase del rango de valores.

InvertirBitEnVariable

Invierte un bit en la variable indicada. Si el bit de la variable presenta el valor 1 (TRUE), se ajusta a 0 (FALSE). Si el bit de la variable tiene el valor 0 (FALSE), se ajusta a 1 (TRUE).

Tras modificar el bit indicado, la función de sistema vuelve a transferir toda la variable al controlador. No se comprueba si, entretanto, se han modificado otros bits en la variable. El operador y el controlador solo podrán acceder a la variable indicada en modo de lectura hasta que la variable vuelva a transferirse al controlador.

Nota

No utilice esta función de sistema si el controlador admite variables Bool. En su lugar, utilice la función de sistema *InvertirBit*.

ActivarBit

Ajusta a 1 (TRUE) el valor de una variable del tipo Bool.

ActivarBitDeVariable

Ajusta un bit a 1 (TRUE) en la variable indicada. Tras modificar el bit indicado, la función de sistema vuelve a transferir toda la variable al controlador. No se comprueba si, entretanto, se han modificado otros bits en la variable. El operador y el controlador solo podrán acceder a la variable indicada en modo de lectura hasta que la variable vuelva a transferirse al controlador.

Nota

No debe utilizarse esta función de sistema si el controlador admite variables Bool. En su lugar, utilice la función de sistema *ActivarBit*.

ActivarBitMientrasTeclaPulsada

Ajusta un bit a 1 (TRUE) en la variable indicada mientras el usuario mantiene pulsada la tecla configurada.

Tras modificar el bit indicado, la función de sistema vuelve a transferir toda la variable al controlador. No se comprueba si, entretanto, se han modificado otros bits en la variable. El operador y el controlador solo podrán acceder a la variable indicada en modo de lectura hasta que la variable vuelva a transferirse al controlador. Para evitar problemas debidos a accesos simultáneos a una misma variable, se recomienda utilizar esta función de sistema solo para acceder a variables del tipo Bool.

Nota
Con el cambio de imagen configurado para una tecla, se ejecutan inmediatamente todas las funciones configuradas para el evento *Soltar*, aunque la tecla todavía permanezca pulsada.

Si, para una tecla de función, se halla configurada la función de sistema *ActivarBitMientrasTeclaPulsada*, el bit se desactivará inmediatamente tras ejecutar el cambio de imagen. Este comportamiento es necesario, puesto que la asignación de las teclas será diferente tras el cambio de imagen.

No debe utilizarse esta función de sistema si el controlador admite variables Bool. En su lugar, utilice la función de sistema *ActivarBit* (11 HMI).

Administración de usuarios

LeerContraseña
Lee la contraseña del usuario conectado al panel de operador y la escribe en la variable indicada.

Nota
Se debe asegurar que el valor de la variable indicada no aparece en ningún otro lugar del proyecto.

Las contraseñas de usuarios de SIMATIC Logon no se pueden leer.

LeerNombreDeUsuario
Escribe el nombre del usuario conectado actualmente al panel de operador en la variable indicada.

Si la variable indicada contiene una conexión con el controlador, el nombre del usuario también estará disponible en el controlador. Con esta función de sistema se puede, por ejemplo, liberar determinadas funciones para usuarios específicos.

LeerNúmeroDeGrupo
Lee el número del grupo al que pertenece el usuario registrado en el panel de operador y lo escribe en la variable indicada.

MostrarDiálogoInicioSesión
Abre un cuadro de diálogo en el panel de operador con el que el usuario puede iniciar una sesión en el panel de operador.

IniciarSesión
Inicia la sesión del usuario actual en el panel de operador.

CerrarSesión
Cierra la sesión del usuario actual en el panel de operador (12 HMI).

Avisos

BorrarBúferDeAvisos
Borra avisos del búfer en el panel de operador.

Nota
Los avisos no acusados también se borran.

MostrarVentanaDeAvisos
Muestra u oculta la ventana de avisos en el panel de operador.

Cálculos

IncrementarVariable
Suma el valor indicado al valor de la variable:

$$X = X + a$$

Nota
La función de sistema utiliza la misma variable como valor de entrada y como valor de salida. Si emplea esta función de sistema para convertir un valor, deberá usar una variable auxiliar. A esta se le puede asignar un valor con la función de sistema *DefinirVariable*.

DecrementarVariable
Resta el valor indicado del valor de la variable:

$$X = X - a$$

Nota
La función de sistema utiliza la misma variable como valor de entrada y como valor de salida. Si utiliza esta función de sistema para convertir un valor, deberá utilizar una variable auxiliar. A la variable auxiliar se le asigna un valor con la función de sistema *DefinirVariable*.

DefinirVariable
Asigna un valor nuevo a la variable indicada.

Nota
Dependiendo del tipo de variable, con esta función de sistema, puede asignar cadenas de caracteres y números.

EscalaLineal
Asigna a la variable Y un valor que se calcula a partir del valor de las variables X indicadas mediante la función lineal $Y = (a * X) + b$.

La función inversa es la función de sistema *InvertirEscalaLineal.*

Nota
Las variables X e Y no pueden ser idénticas. Si desea convertir una variable en sí misma, debe utilizar una variable auxiliar.

Con la función de sistema *DefinirVariable,* se puede asignar a la variable auxiliar el valor de la variable que se va a convertir.

InvertirEscalaLineal
Asigna a la variable X un valor calculado a partir del valor de la variable Y indicada mediante la función lineal $X = (Y\ b) / a$.

Las variables X e Y no pueden ser idénticas. Esta función de sistema es la función inversa de *EscalaLineal.*

Nota
Las variables X e Y no pueden ser idénticas. Si desea convertir una variable en a sí misma, debe utilizar una variable auxiliar.

Con la función de sistema *DefinirVariable,* se puede asignar a la variable auxiliar el valor de la variable que se va a convertir.

Configuración

DefinirFechaHoraPLC
Modifica la fecha y la hora del controlador conectado.

La función de sistema *AjustarFechaHoraPLC* solo se puede configurar para los siguientes controladores: SIMATIC S7 1200 y SIMATIC S7 1500.

CambiarConexión
Deshace la conexión con el controlador que se está utilizando y establece una conexión con un controlador en una dirección distinta. El controlador recién conectado debe pertenecer a la misma clase de dispositivos (S7-1200 o S7-300). Además, en el S7-1200, la utilización de la función solo resulta admisible con direccionamiento absoluto.

Nota
Al cambiar a una dirección diferente, vigile que esta dirección no esté siendo utilizada por otro panel de operador. Se admiten las direcciones IP y MPI.

DefinirModoPLC
Cambia el modo de operación del controlador a uno de los siguientes estados: RUN – STOP. La función de sistema *DefinirModoPLC* solo se puede configurar para los siguientes controladores: SIMATIC S7 1200 y SIMATIC S7 1500.

EstablecerIdioma
Cambia el idioma del panel de operador. Todos los textos y avisos del sistema configurados aparecerán en el panel de operador en el idioma recién ajustado (19 HMI).

EstablecerModoDeConexión
Establece o desconecta la conexión indicada.

Nota
Solo puede establecerse una conexión con el controlador si el panel de operador funciona en modo *online*. Utilice, para ello, la función de sistema *EstablecerModoDeOperaciónPanel*.

EstablecerModoDeOperaciónPanel
Cambia el modo de operación en el panel de operador. Se dispone de los modos de operación siguientes: *Online*, *Offline* y *Cargar*.

LeerModoPLC
Evalúa el estado actual del controlador conectado. La función de sistema *LeerModoPLC* solo se puede configurar para los siguientes controladores: SIMATIC S7 1200 y SIMATIC S7 1500.

Ficheros

AbrirTodosLosFicheros
Restablece la conexión entre WinCC y los ficheros. La archivación puede continuar.

Nota
Para reiniciar el proceso de archivar, ejecute la función de sistema *IniciarArchivación*.

ArchivarVariable
Guarda el valor de la variable indicada en el fichero de variables deseado. Se debe utilizar esta función de sistema si desea archivar un valor de proceso en un momento determinado.

BorrarFichero:
Borra todos los registros del fichero indicado.

CerrarTodosLosFicheros
Deshace la conexión entre WinCC y todos los ficheros.

Nota
Antes de cerrar un fichero, se ha de detener la archivación en dicho fichero. Utilice, para ello, la función de sistema *PararArchivar*.

IniciarArchivación:
Inicia la archivación de valores de proceso o avisos en el fichero indicado. La función también puede aplicarse a los Audit Trails.

La archivación se puede cancelar en *runtime* si se utiliza la función de sistema *PararArchivación*.

PararArchivación:
Detiene la archivación de valores de proceso o avisos en el fichero indicado. La función también puede aplicarse a los Audit Trails. La archivación se puede reiniciar en runtime utilizando la función de sistema *IniciarArchivación*.

Nota
Cuando se detiene la archivación, permanece establecida la conexión entre WinCC y los archivos comprimidos o la base de datos de ficheros. Para deshacer dicha conexión, utilice la función de sistema *CerrarTodosLosFicheros*.

Imágenes

ActivarImagen
Efectúa un cambio de imagen con el objetivo de visualizar la imagen indicada. Para conmutar entre la imagen raíz y la ventana permanente, y viceversa, utilice la función de sistema *ActivarImagenConNúmero*.

ActivarImagenAnterior
Torna a la imagen que estaba activada antes de cambiar a la imagen actual. La modificación de imagen no se ejecuta si antes no había ninguna imagen activa. Se guardan las últimas 10 imágenes activadas. Si cambia a una imagen que ya no estuviera guardada, aparecerá un aviso del sistema.

Nota
Si desea utilizar la función de sistema, la imagen a la que desea cambiar deberá estar contenida en la estructura de navegación.

ActivarImagenConNúmero
Dependiendo del valor de una variable, cambia a una imagen determinada. La imagen se identifica por su número de imagen.

Manejo del teclado para objetos de imágenes

NavegadorHTMLDesplazarIzquierda
Pasa página a la izquierda en el navegador HTML.

NavegadorHTMLAcercar
Aumenta la representación del navegador HTML en un grado de *zoom*.

NavegadorHTMLActualizar
Ejecuta la función *Actualizar* del navegador HTML.

NavegadorHTMLAdelante
Ejecuta la función *Adelante* del navegador HTML.

NavegadorHTMLDesplazarDerecha
Pasa página a la derecha en el navegador HTML.

NavegadorHTMLAlejar
Reduce la representación del navegador HTML en un grado de *zoom*.

NavegadorHTMLAtrás
Ejecuta la función Atrás del navegador HTML.

NavegadorHTMLAvanzar
Avanza página en el navegador HTML.

NavegadorHTMLCancelar
Ejecuta la función *Cancelar* del navegador HTML.

NavegadorHTMLPáginaAbajo
Hace que el cursor se desplace por páginas hacia abajo en el navegador HTML (26 HMI).

NavegadorHTMLPáginaArriba
Hace que el cursor se desplace por páginas hacia arriba en el navegador HTML.

NavegadorHTMLPáginaInicial
Pasa a la página inicial almacenada para el navegador HTML.

NavegadorHTMLRetroceder
Retrocede página en el navegador HTML.

ObjetoDeImagenRetrocederPágina
Hace que el cursor se desplace por páginas hacia arriba en el objeto de imagen indicado. Esta función de sistema se puede utilizar con los objetos de imagen indicados a continuación:

- Visor de usuarios.
- Visor de avisos.
- Visor de recetas.
- Visor de parte del programa NC.

ObjetoDeImagenCursorAbajo
Hace que el cursor se desplace por líneas hacia abajo en el objeto de imagen indicado.

Esta función de sistema se puede utilizar con los objetos de imagen indicados a continuación:

- Visor de usuarios.
- Visor de avisos.
- Visor de recetas.
- Visor de parte de programa NC.

ObjetoDeImagenCursorArriba
Hace que el cursor se desplace por líneas hacia arriba en el objeto de imagen indicado.

Esta función de sistema se puede utilizar con los objetos de imagen indicados a continuación:

- Visor de usuarios.
- Visor de avisos.
- Visor de recetas.
- Visor de parte del programa NC.

ObjetoDeImagenAvanzarPágina
Hace que el cursor se desplace por páginas hacia abajo en el objeto de imagen indicado.

Esta función de sistema se puede utilizar con los objetos de imagen indicados a continuación:

- Visor de usuarios.
- Visor de avisos.
- Visor de recetas.
- Visor de parte de programa NC.

ObjetoDeImagenCursorDerecha
Hace que el cursor se desplace por líneas hacia la derecha en el objeto de imagen indicado.

ObjetoDeImagenCursorIzquierda
Hace que el cursor se desplace por líneas hacia la izquierda en el objeto de imagen indicado.

VisorDeAvisosAcusarAviso
Acusa los avisos seleccionados en el visor de avisos indicado. Se utiliza esta función de sistema si no se debe utilizar el botón integrado en el control ActiveX.

VisorDeAvisosLoopInAlarm
Lanza el evento *Loop-In-Alarm* para todos los avisos seleccionados en el visor de avisos indicado.

Se utiliza esta función de sistema si no se debe utilizar el botón integrado en el control *ActiveX*.

Puede configurar, a su vez, una función de sistema para el evento *Loop-In-Alarm;* por ejemplo, puede conmutar a la imagen del proceso en la que se ha presentado el aviso.

Nota
Si no ha acusado todavía los avisos para editar, el acuse se efectuará automáticamente al ejecutarse esta función de sistema.

VisorDeAvisosMostrarTextoAyuda
Muestra el *tooltip* configurado para el aviso seleccionado en el visor de avisos indicado.

VisorDeCurvasActivarModoRegla
Muestra u oculta la regla en el visor de curvas indicado. La regla muestra el valor Y correspondiente a un valor X.

Nota
Para visualizar la regla, se debe activar el ajuste *Mostrar regla* en las propiedades del visor de curvas.

VisorDeCurvasAmpliar
Reduce el intervalo de tiempo que se visualiza en el visor de curvas indicado.

VisorDeCurvasAvanzarPágina
Avanza un ancho de visualización en el visor de curvas indicado.

VisorDeCurvasComprimir
Aumenta el intervalo de tiempo que se visualiza en el visor de curvas indicado.

VisorDeCurvasIniciarPara
Detiene o prosigue el registro de curvas en el visor de curvas indicado.

VisorDeCurvasIrAlPrincipio
Retrocede al principio del área de representación en el visor de curvas.

VisorDeCurvasReglaDerecha
Desplaza la regla hacia delante (a la derecha) en el visor de curvas indicado.

Nota
Para poder desplazar la regla, esta debe estar activada. Para activarla utilice la función de sistema *VisorDeCurvasActivarModoRegla.*

VisorDeCurvasReglaIzquierda
Desplaza la regla hacia atrás (a la izquierda) en el visor de curvas indicado.

Nota
Para poder desplazar la regla, esta debe estar activada. Para activarla, utilice la función de sistema *VisorDeCurvasActivarModoRegla.*

VisorDeCurvasRetrocederPágina
Retrocede un ancho de visualización en el visor de curvas indicado.

VisorDeRecetasAbrir
Muestra los valores de registro en el visor de recetas indicado o cambia al siguiente campo de selección. Esta función de sistema no posee ningún efecto si el campo de selección de los valores del registro de receta se muestra en el panel de operador.

El orden de manejo de los campos de selección en *runtime* es el siguiente:

- Nombre de receta.
- Nombre de registro.
- Valores de registros de recetas.

Utilice esta función de sistema si ha configurado el visor de recetas simple. En este se muestra, en cada caso, un solo campo de selección en el panel de operador. Utilice la función de sistema *VisorDeRecetasAtrás* para ver el campo de selección anterior.

VisorDeRecetasAtrás
Vuelve al campo de selección anterior en el visor de recetas indicado.

Esta función de sistema no posee ningún efecto si el campo de selección de la receta se muestra en el panel de operador.

Orden de manejo de los campos de selección en runtime:

- Nombre de receta.
- Nombre de registro.
- Valores de registros de recetas.

Utilice esta función de sistema si ha configurado el visor de recetas simple. En este se muestra, en cada caso, un solo campo de selección en el panel de operador. Se utiliza la función de sistema *VisorDeRecetasAtrás* para ver el campo de selección anterior.

VisorDeRecetasBorrarRegistro
Cambia el nombre del registro seleccionado en el visor de recetas indicado.

VisorDeRecetasCambiarNombreDeRegistro
Transfiere al controlador el registro de receta visualizado actualmente en el visor de recetas.

VisorDeRecetasGuardarRegistro
Guarda el registro de receta visualizado actualmente en el visor de recetas.

VisorDeRecetasGuardarRegistroComo
Guarda con otro nombre el registro que se visualiza en el visor de recetas.

VisorDeRecetasLeerRegistroDelControlador
Transfiere el registro cargado en el controlador al panel de operador y lo muestra en el visor de recetas.

VisorDeRecetasMenú
Abre el menú del visor de recetas simple indicado.

Utilice esta función de sistema solo en un visor de recetas simple.

VisorDeRecetasMostrarTextoDeAyuda
Muestra el tooltip configurado para el visor de recetas indicado.

VisorDeRecetasRegistroNuevo
Crea un registro nuevo en el visor de recetas visualizado.

VisorDeRecetasSincronizarRegistroConVariables
Sincroniza los valores del registro que se visualiza en el visor de recetas con las variables de receta correspondientes. Se utiliza esta función de sistema solo en el visor de recetas avanzado.

Durante la sincronización, todos los valores del registro se escriben en las variables de receta correspondientes.

VisorDiagnósticoSistemaActualizarBúferPLC
Actualiza los datos del búfer de diagnóstico del controlador en el visor de diagnóstico del sistema.

VisorDiagnósticoSistemaAtrás
Pasa a la vista de orden superior en un diagnóstico de sistema:

- Si en el visor de diagnóstico del sistema se representa la vista del búfer de diagnóstico, la función abre la vista de dispositivos.
- Si en el visor de diagnóstico del sistema se representa la vista detallada, la función abre la vista del búfer de diagnóstico.

VisorDiagnósticoSistemaVistaDetallada
Abre la vista detallada en un diagnóstico de sistema para mostrar información detallada sobre la conexión seleccionada y los errores existentes.

Otras funciones

ActualizarVariable:
Lee del controlador el valor actual de la variable con la ID de actualización indicada.

PararRuntime:
Sale del *software* runtime y, con ello, del proyecto que se está ejecutando en el panel de operador.

20. HMI Avisos

Los avisos son llamadas de atención que se producen cuando concurren determinados factores, unos programados por el usuario y otros generados por el propio sistema. Dichos avisos se visualizan en la pantalla.

Dentro de los avisos que pueden ser definidos por el usuario, se encuentran los siguientes tipos:

- Avisos analógicos.
- Avisos de tipo bit.
- Avisos de control.
- Avisos de usuario.

Estos dos últimos tipos no se encuentran disponibles en todos los paneles de operador. Concretamente, los paneles básicos no los presentan (como en la pantalla del ejemplo, KTP700).

En este ejercicio se van a realizar avisos analógicos y de bit. Se comenzará por los últimos.

Avisos de bit

Para generar avisos de tipo bit, se debe preparar el OB1 y, posteriormente, se configura la gestión de los avisos en WinCC. Para gestionar los avisos en WinCC, se ha de hacer a través de una variable de tipo entera o Word. Y, para ello, se utilizan datos de un DB. Otra condición importante es que el tratamiento del DB sea mediante direccionamiento absoluto y no simbólico; es decir, que se pueda acceder a los datos del DB mediante sus direcciones de tipo DB1.DBX3.0 (absoluto). Por defecto, todos los datos de bloques se gestionan con direccionamiento simbólico y no permiten direccionamiento absoluto, por lo que habrá que modificarlos en sus propiedades. En ese bloque de datos creado, se colocarán todas las alarmas o avisos. En este ejercicio se va a utilizar un grupo de entradas que activarán diversos avisos. Cuando alguna de esas entradas se active, en la pantalla surgirán los avisos correspondientes. En la siguiente lista aparecen las entradas junto con el aviso asociado:

E0.0 Aviso de *Activación de fallo de llenado*.

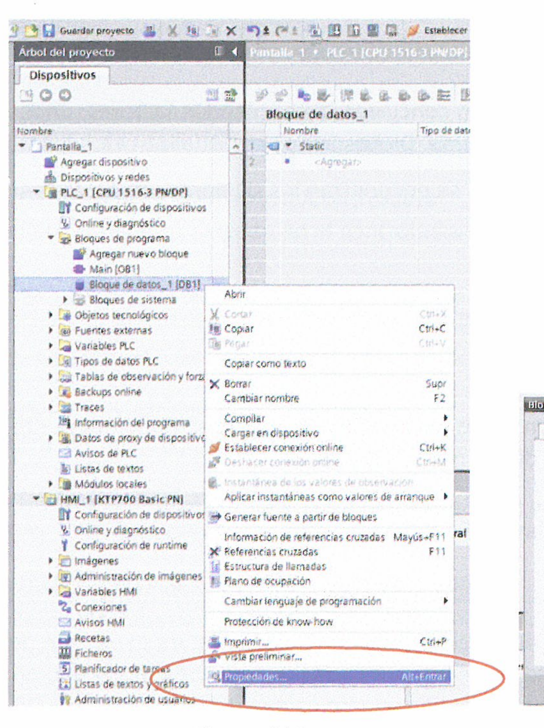
E0.1 Aviso de *Activación de fallo de vaciado*.

E0.2 Aviso de *Activación Cilindro bloqueado*.

Se va a utilizar un DB, al que se le van a añadir esas tres alarmas mediante tres variables de bit.

Preparación del PLC

Se comienza por la creación del DB. Se añade un nuevo bloque, un DB en este caso, que, por defecto, será global. Una vez creado, con el botón derecho tocando en el bloque, se accede a sus propiedades. En atributos se debe quitar la única marca que hay en *Acceso optimizado al bloque* (figuras 282-283).



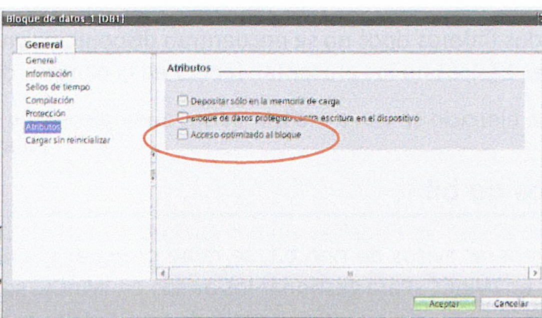


Figura 282

Figura 283

En el DB1 se añaden tres datos de tipo bit, como en la figura 284.

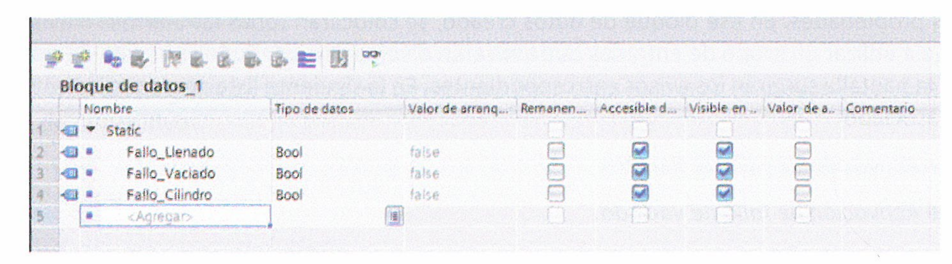


Figura 284

Una vez que se transmita el DB1 al autómata, en la columna *Offset* aparecerán las direcciones que tienen los datos dentro del DB1 (figura 285).

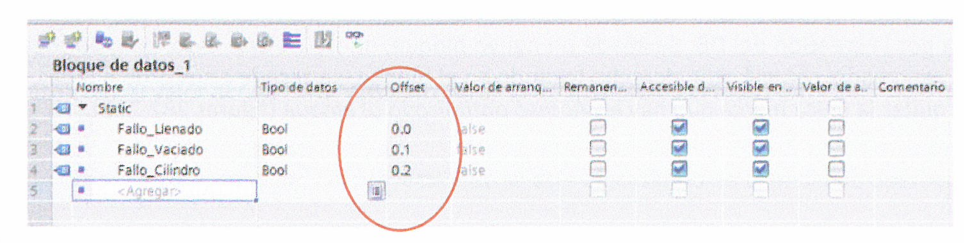
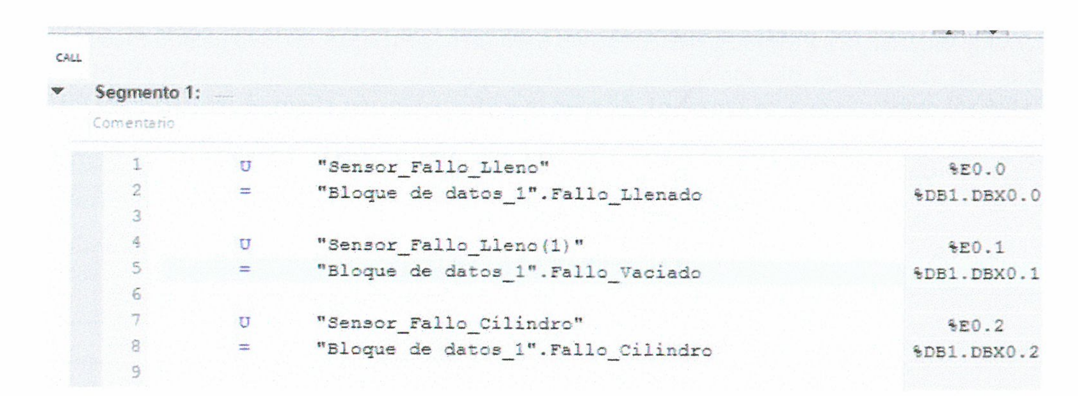


Figura 285

Para acceder al dato del DB1 *Fallo_Llenado,* se accede con la siguiente dirección: DB1.DBX0.0; para acceder al dato *Fallo_Vaciado*, se hace mediante la dirección DB1.DBX0.1 y, para acceder a *Fallo_Cilindro*, se utiliza la dirección DB1.DBX0.2.

El programa del OB1 para generar las alarmas es el siguiente:



Configuración de la pantalla. Variables de alarmas

Ahora se debe preparar la pantalla para que los avisos se generen y aparezcan en la pantalla como una alarma. Existen varios niveles de «importancia» en los avisos y estos pueden necesitar o no. Si fuese necesario acusar aviso, habría que accionar un botón de la pantalla para que el aviso desaparezca.

Se va a empezar creando la lista de los avisos. Para ello, desde el árbol del proyecto, se accede a *Avisos HMI* y, una vez abierto el editor de avisos, se abre la pestaña de los *Avisos de bit* (figura 286).

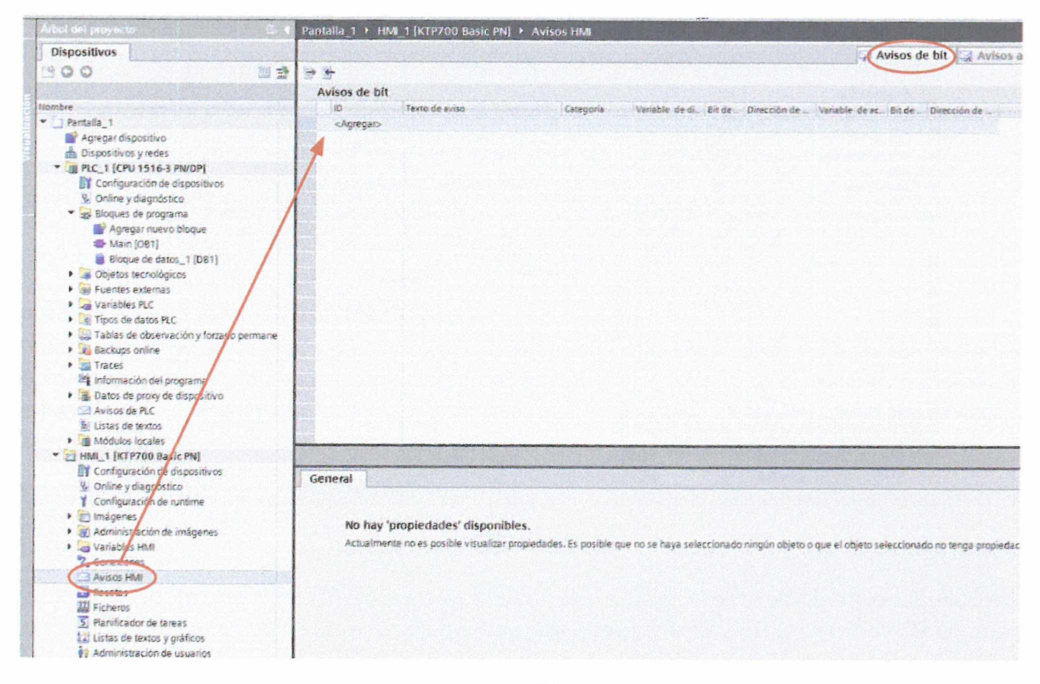


Figura 286

Se dispone de varios campos para completar los avisos. En el *Texto de aviso,* se pone lo que se desea que salga en la pantalla cuando salte la alarma. En *Categoría* se selecciona si se trata de un *error* o de un *warning*. Si se coloca como error, se deberá acusar, porque se considera una alarma de un nivel más importante. Si se selecciona como *warning*, constituye un aviso con un nivel menos importante y no hace falta acuse de recibo. En el error, una vez que se produzca la alarma, ya no desaparecerá el aviso, aunque sí lo haga el hecho físico que lo ha provocado. En el caso de ser un *warning*, sí que se eliminará el aviso cuando desaparezca el hecho que lo haya ocasionado.

En *Variable alarmas,* se debe crear una variable nueva como variable de las alarmas. Como se ha dicho, ha de tratarse de una variable de tipo Word perteneciente al DB1 y debe contener las variables ya configuradas en el DB1, por lo que la nueva variable que se creará será la DB1.DBW0.

Para ello se pulsa sobre los **puntos suspensivos** para agregar una nueva variable, como se detalla en la figura 287.

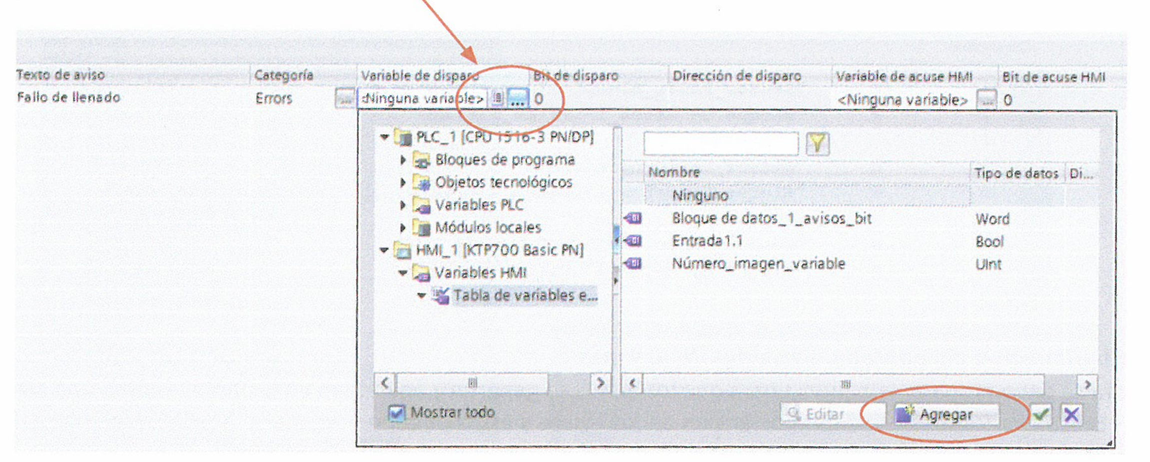


Figura 287

Se rellenan los campos para crear la variable de avisos deseada. Se pone nombre a la variable (aquí *Variable alarmas*) y se selecciona la conexión de HMI (figura 288).

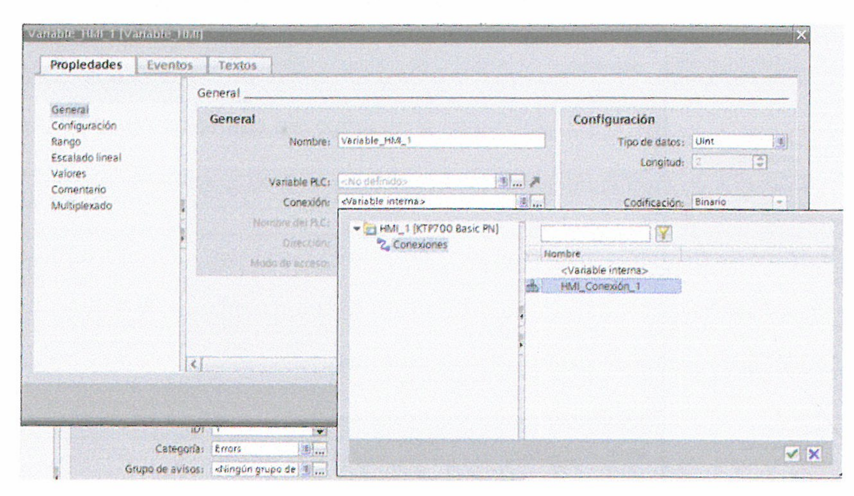


Figura 288

Se debe seleccionar el modo de acceso del DB, que ha de ser absoluto. Al elegir este modo de acceso, se habrá colocado un DB como dirección, pero hay que asegurarse de que sea la correcta. En este caso, la correcta es la DB1.DBW0. En la figura 289 se indica tal situación.

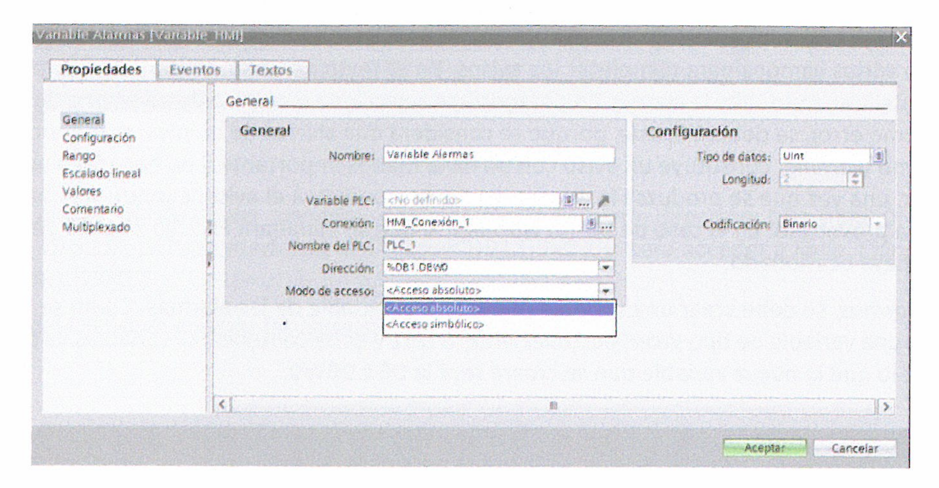
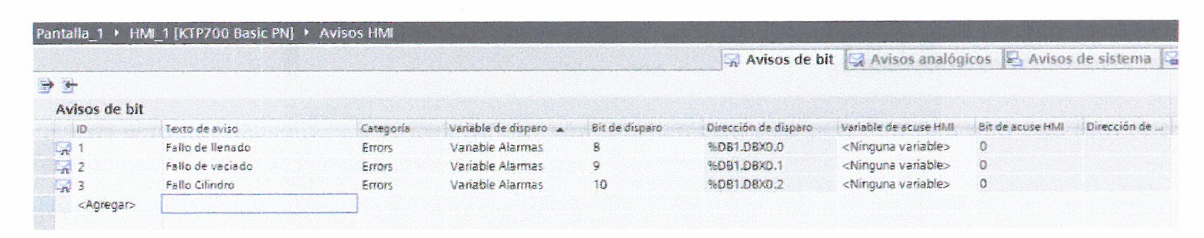


Figura 289

El siguiente dato que se debe introducir es el número de bit del aviso dentro de la variable de avisos (DB1.DBW0). Se ha de poner el bit del aviso que ocupa dentro del DB1. Para ello, se recuerda que la dirección DB1.DBW0 se encuentra formada por el *byte* B0 y el *byte* B1. La forma en que se cargan los *bytes* en una palabra es la siguiente: el *byte* bajo, a la izquierda y el *byte* alto, a la derecha. Como los bits de aviso que anteriormente se han escogido en el DB eran el bit 0 *(Fallo_Llenado)*, bit 1 *(Fallo_Vaciado)* y bit 2 *(Fallo_Cilindro)* y pertenecían al *byte* 0 (*byte* bajo), se situarán en la posición 8, 9 y 10 de la palabra. Puede verse en el siguiente gráfico.



B0								B1							
B0.7	B0.6	B0.5	B0.4	B0.3	B0.2	B0.1	B0.0	B1.7	B1.6	B1.5	B1.4	B1.3	B1.2	B1.1	B1.0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits de los avisos en el DB1

Ahora falta por especificar la dirección absoluta de la alarma en el DB1 que, para el primer aviso, será DB1.DBX0.0.

En la figura 290 se indican los tres avisos configurados.

Figura 290

Configuración de la pantalla. Visor de alarmas

Ya están preparadas las variables, tanto en el PLC como en la pantalla de HMI. Falta definir el método de la visualización de los avisos en la pantalla. Existen dos posibles métodos. Uno es mediante la pantalla general. Si se ha utilizado el asistente para seleccionar la pantalla, se habrá creado una pantalla general con tres visores de avisos, como se observa en la figura 291.

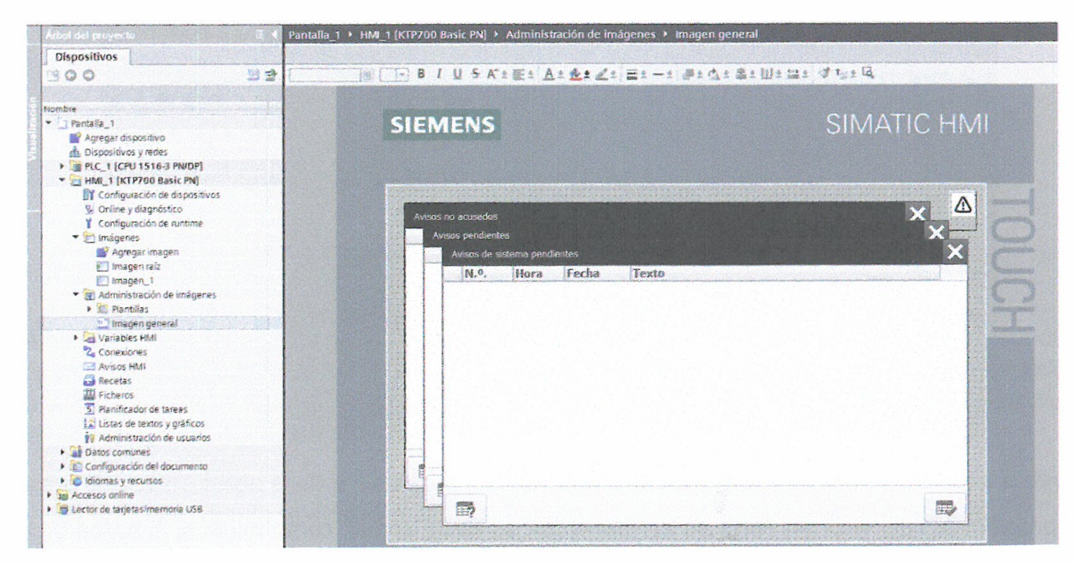


Figura 291

Dos visores son de avisos de usuario y otro de avisos del sistema. En los de usuario, se dispone de dos ventanas: una en la que aparecen los avisos mientras este dura, *Avisos pendientes*, y otra en la que el aviso queda hasta que se ha acusado (enterado), aunque este haya desaparecido. También se dispone de un icono (⚠), que indica el número de avisos que hay disparados; pulsando sobre él, se pueden hacer aparecer y desaparecer los avisos pendientes. Para seguir el mismo procedimiento con el visor de alarmas no acusadas, se pulsa sobre el icono ⚠. Para acusar se presiona en el icono . Este método resulta el más rápido y cuenta con otra ventaja, que es que aparece en cualquier imagen; en cuanto se produce la alarma, salta la ventana de avisos.

En este método se necesita activar la opción de visualización de los *warnings*. Esto se realiza en las propiedades de los visores de avisos en *Imagen general*, como se indica en la figura 292.

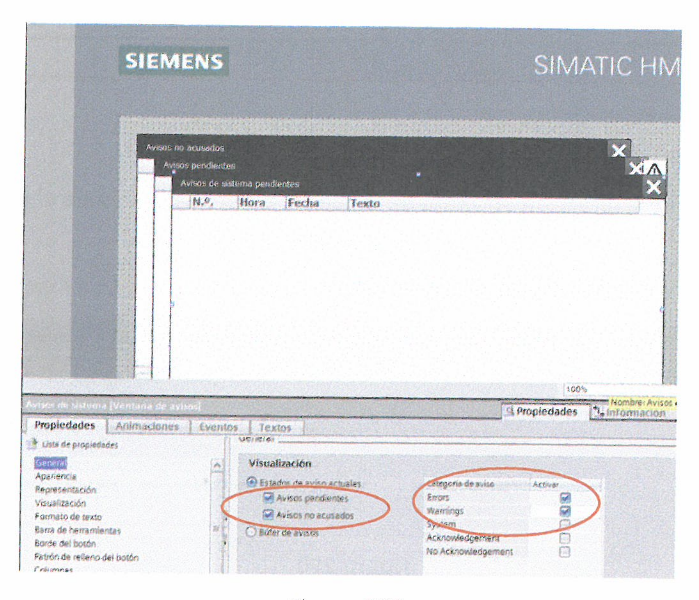


Figura 292

El otro método consiste en el empleo del visor del menú de *Controles*, a la derecha, en las *Herramientas*. Este visor se coloca en una imagen concreta, pero solo se verá si se encuentra en esa ventana cuando salten las alarmas. Se puede también hacer que «salte», pero siempre resulta más rápido y cómodo el sistema anterior. Se pueden combinar igualmente los dos juntos. En la figura 293 se muestra el visor de avisos.

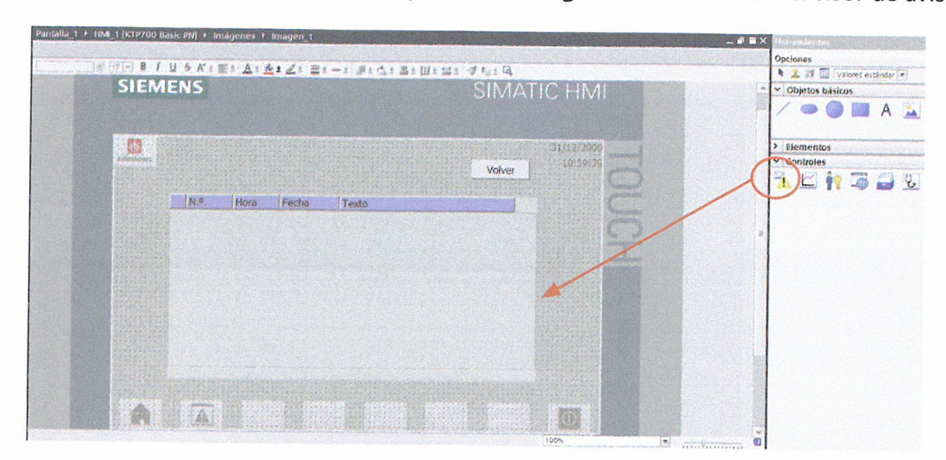


Figura 293

En las propiedades del visor en *Barra de herramientas*, se ha de añadir, como mínimo, el botón de acuse (figura 294).

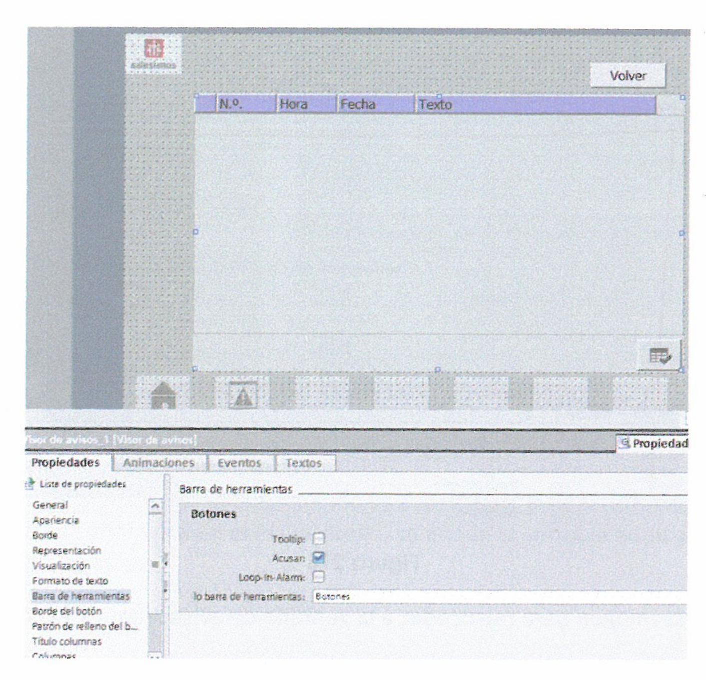


Figura 294

Para ver el comportamiento de los *warnings*, se va a cambiar uno de los avisos: *error* a *warning*. El aviso de *Fallo de vaciado* se sitúa como *warning*.

Hay que fijarse en que el aviso de *warning* desaparece del visor en cuanto deja de existir. No ocurre lo mismo con las alarmas. En el visor, a las alarmas se les añade una exclamación (!) (figura 295).



Figura 295

Si se hubiera marcado también la opción *Loop in alarm*, se hubiera podido acceder a una imagen donde ofrecer información sobre la alarma. Se va a asociar una nueva imagen, donde se insertará un texto explicativo para la alarma de *Fallo llenado*. El texto será: «CUIDADO!!!! EL DEPÓSITO ESTÁ LLENO Y LA ELECTROVÁLVULA NO CIERRA». Saldrá en rojo e intermitente.

El proceso debe ser: desde el visor de avisos y en las propiedades del aviso que nos interesa, se indica el nombre de la imagen donde se dirigirá al activar el botón del *Loop in alarma* ( ) (figura 296).

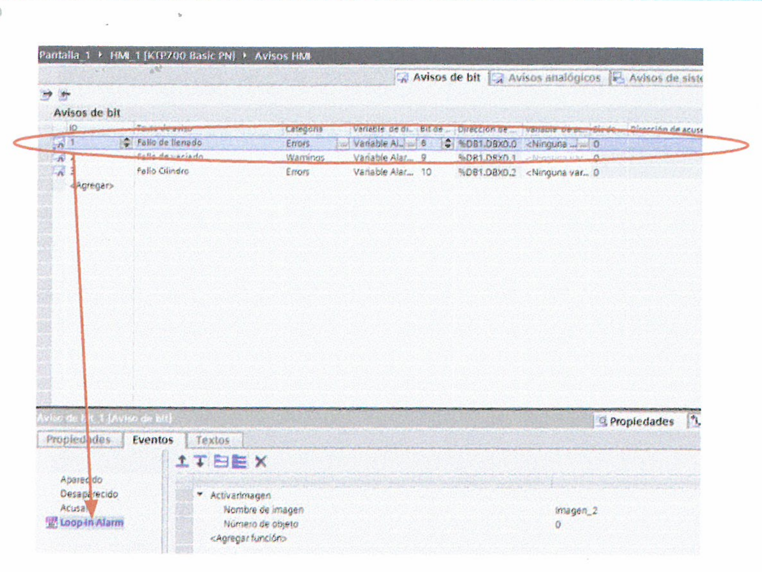


Figura 296

Se añade un botón en esa imagen para retroceder a la ventana de avisos (figura 297).

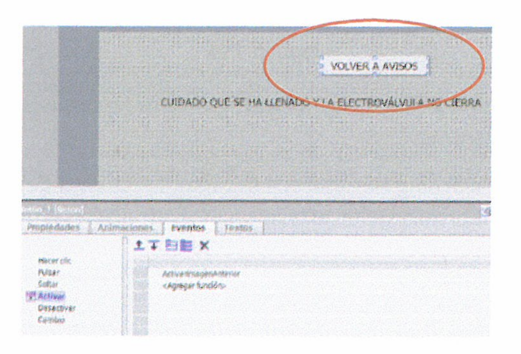


Figura 297

Se marca la opción *Loop in alarm* en las propiedades del visor (figura 298).

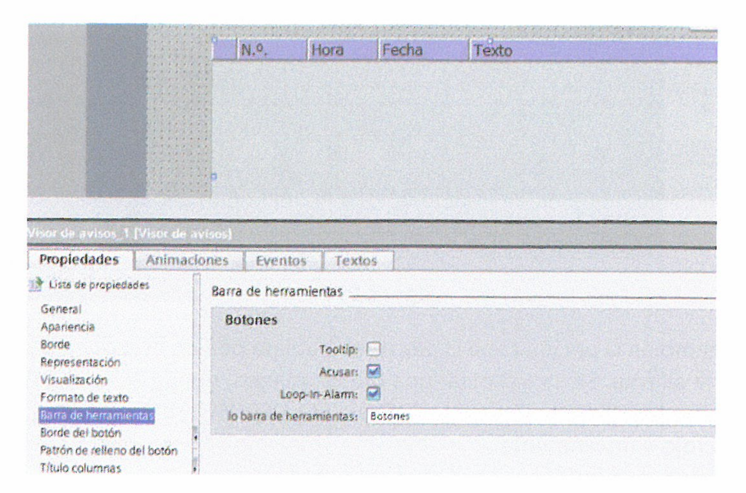


Figura 298

Se añade una nueva imagen donde se pondrá el texto referido anteriormente. En las propiedades del texto, se señala el color rojo e intermitente (figuras 299-300).

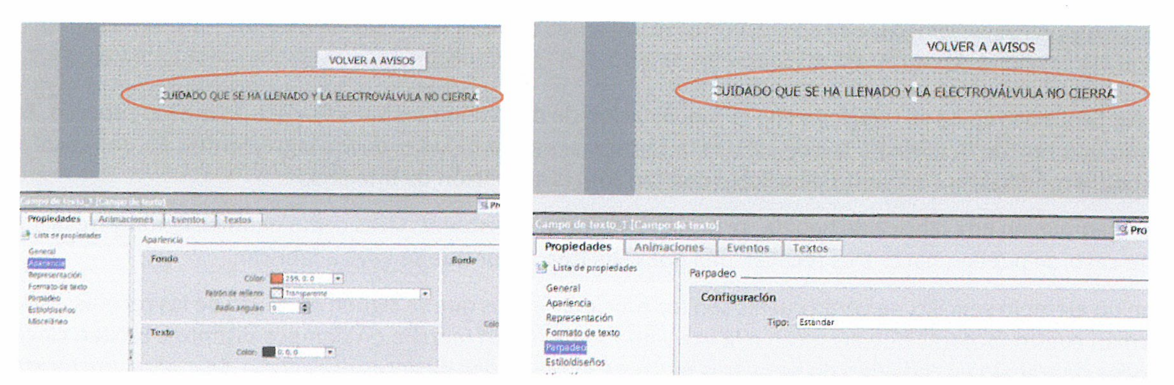
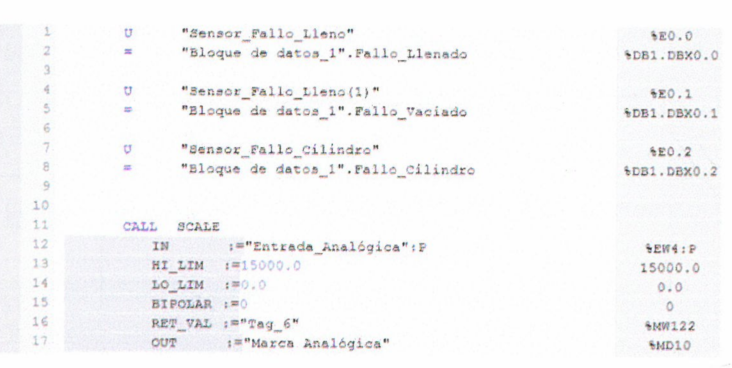


Figura 299 Figura 300

Configuración de la pantalla. Avisos analógicos

Se va a añadir un aviso analógico, de forma que, cuando se esté cerca del llenado, avise con un *warning*. Para ello se debe tratar la entrada analógica en el PLC y, luego, se añade la variable analógica y el valor que hace disparar el aviso en la lista de avisos.

En el OB1 del autómata se debe añadir lo siguiente:



La entrada analógica presenta dirección 4 y se guarda en la marca de palabra 10. Los límites están entre 0 y 15.000 litros. El valor del aviso se produce cuando llega a 14.000 litros.

En la lista de aviso de HMI se añade la variable analógica (figura 301).

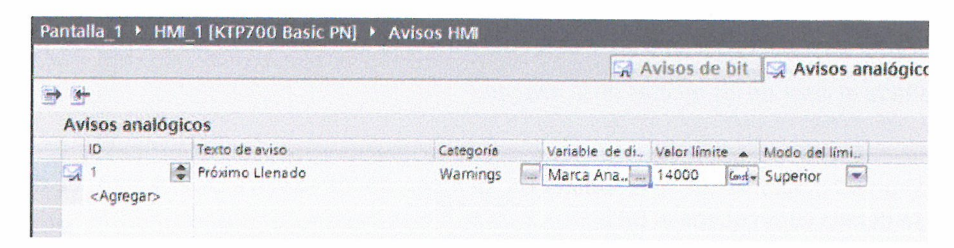


Figura 301

21. HMI Recetas

El concepto de receta se relaciona con la modificación de determinados valores para un mismo proceso. Es como una receta de cocina a la que solo se le cambiaran las cantidades de los ingredientes. En una receta, se podrá modificar un conjunto de valores con tan solo la activación de una tecla.

Por ejemplo, en una fábrica de bebida, se preparan diferentes sabores, tales como naranja, limón y fresa. En este caso, se pueden hacer tres recetas, una por cada tipo sabor que se vaya a fabricar.

Con un ejemplo sencillo, se verán cuáles son los procedimientos que se siguen en el uso de las recetas. Conforman cuatro motores que arrancan a la vez y que se hallan en marcha en tiempos distintos. Existen cinco modos de trabajo en cada uno de estos tres motores, con tiempos distintos de marcha. Se creará una receta para que, desde la propia pantalla de HMI, se seleccione el régimen de trabajo de estos cuatro motores. En la siguiente tabla se indican los modos de trabajo y sus características:

	Motor_1	Motor_2	Motor_3	Motor_4
MODO 1	1 s	2 s	3 s	4 s
MODO 2	2 s	6 s	14 s	30 s
MODO 3	3 s	5 s	7 s	9 s
MODO 4	5 s	7 s	9 s	11 s
MODO 5	14 s	18 s	22 s	32 s

El proceso de trabajo será el siguiente:

1. Se crea un DB para introducir los cuatro tiempos.
2. Se elabora el programa de arranque de los temporizadores.
3. Se diseña la receta con la opción *Recetas* de WinCC, lo que conforma sus elementos y registros.
4. Se añade el visor de las recetas en la imagen.

1. Se va a crear un *DB global,* al que se le llamará *DB_tiempos*, donde se van a crear 4 datos de tiempo. En la figura 302 se detalla cómo queda el DB1.

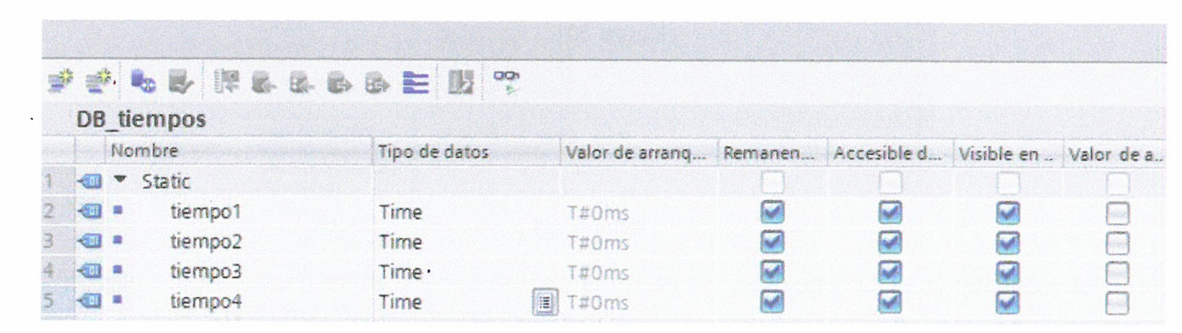


Figura 302

2. Ahora, en el OB1, se van a arrancar 4 temporizadores IEC del tipo TP. Este tipo de temporizador permanece activado durante el tiempo de temporización. Todos los temporizadores arrancan con la misma entrada E1.7 *(Arranque)*. El programa es el siguiente:

```
CALL  TP ,  "IEC_Timer_0_DB_7"           %DB7
    Time
    IN  :="Arranque"                     %E1.7
    PT  :="DB_tiempos".tiempo1
    Q   :="Motor_1"                      %A1.0
    ET  :=

CALL  TP ,  "IEC_Timer_0_DB_8"           %DB11
    Time
    IN  :="Arranque"                     %E1.7
    PT  :="DB_tiempos".tiempo2
    Q   :="Motor_2"                      %A1.1
    ET  :=
CALL  TP ,  "IEC_Timer_0_DB_9"           %DB12
    Time
    IN  :="Arranque"                     %E1.7
    PT  :="DB_tiempos".tiempo3
    Q   :="Motor_3"                      %A1.2
    ET  :=

CALL  TP ,  "IEC_Timer_0_DB_10"          %DB13
    Time
    IN  :="Arranque"                     %E1.7
    PT  :="DB_tiempos".tiempo4
    Q   :="Motor_4"                      %A1.3
    ET  :=
```

3. Desde el árbol del menú, se abre la opción *Receta* para crearla. Aparece en la ventana principal la pantalla de la figura 303.

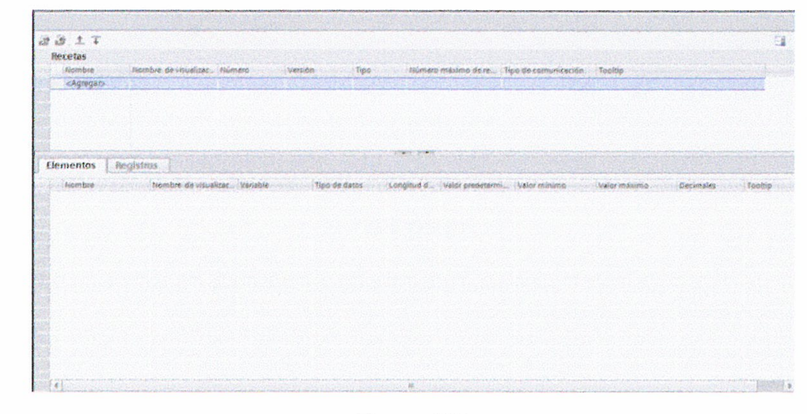


Figura 303

Lo primero que se hará es ponerle un nombre a la receta y el nombre que se desea que aparezca en la pantalla. A continuación, se irán añadiendo los distintos elementos de la receta que, en este caso, se corresponden con los cuatro tiempos. Se añade el nombre del elemento y el nombre que se desea que aparezca en la pantalla. Luego, se asigna una variable a dicho elemento. La variable habrá que seleccionarla del DB creado, tras haberla buscado en el menú que aparece cuando pulsamos en la columna de *Variable*, como se ve en la figura 304.

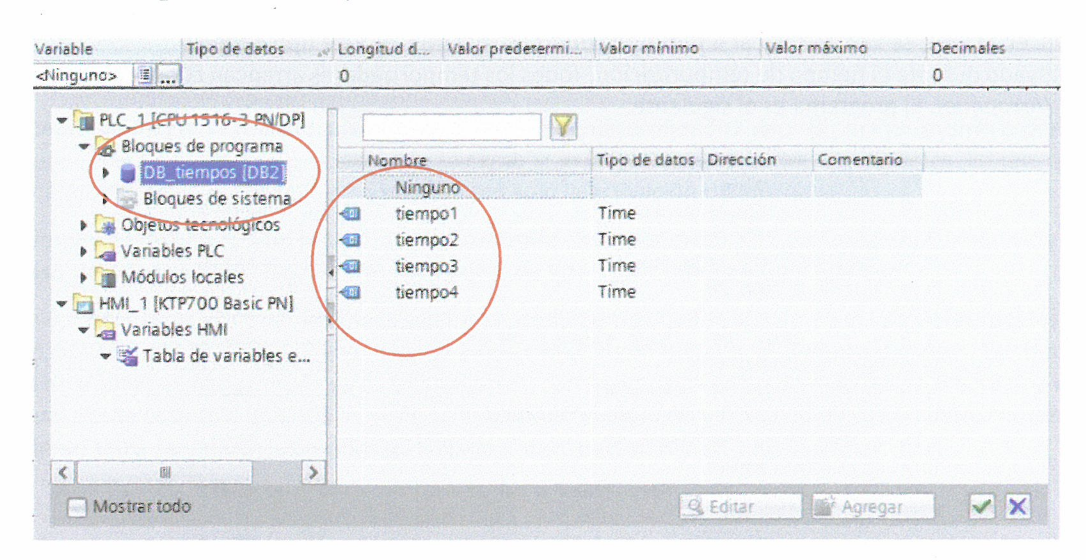


Figura 304

El resto de columnas de los elementos se pueden dejar como salen por defecto. En la figura 305 se distinguen los cuatro tiempos de los elementos.

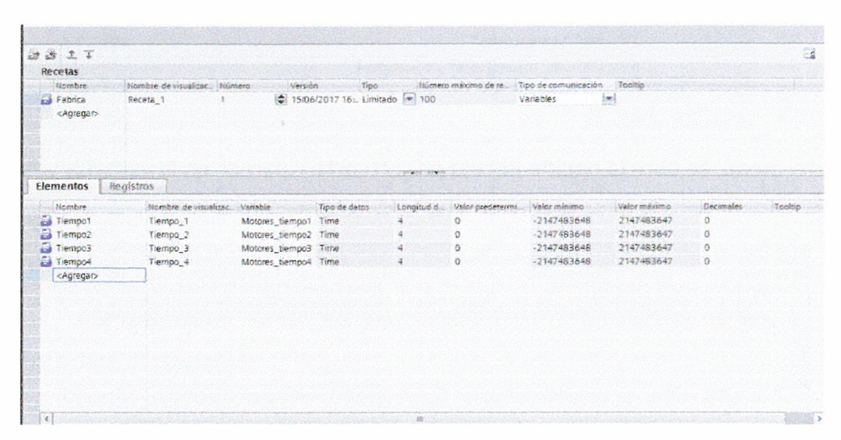


Figura 305

En la pestaña de *Registros,* se añaden los cuatro modos con los tiempos de cada uno de ellos. Los tiempos deben introducirse en milisegundos (figura 306).

Elementos | Registros

Nombre	Nombre de visualizac...	Número	Tiempo1	Tiempo2	Tiempo3	Tiempo4
MODO1	MODO1	1	1000	2000	3000	4000
MODO2	MODO2	2	2000	6000	14000	30000
MODO3	MODO3	3	3000	5000	7000	9000
MODO4	MODO4	4	5000	7000	9000	11000
MODO5	MODO5	5	14000	18000	22000	32000

Figura 306

4. Por último, se debe añadir el visor que aparecerá en la imagen. Para ello, se va a la parte derecha en *Herramientas* y, en la carpeta *Controles*, se arrastra desde el *Visor de recetas* (figura 307).

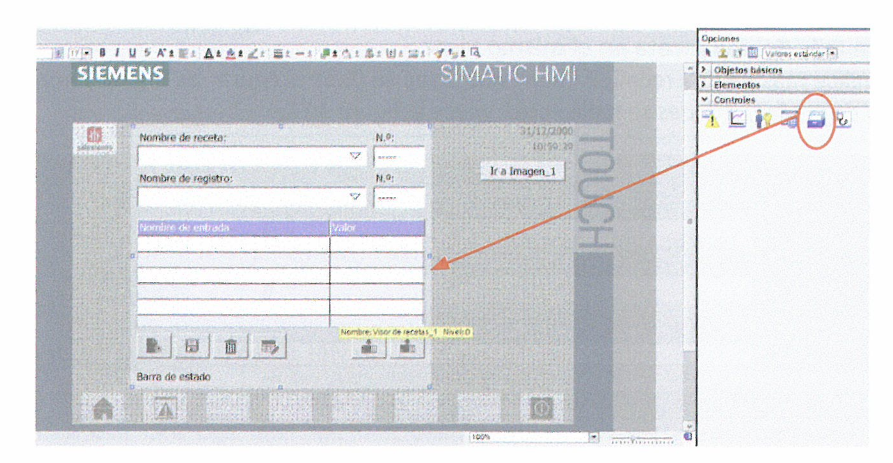


Figura 307

Una vez hecho todo esto, solo falta enviarlo al PLC y la pantalla de HMI. La imagen de la receta es la que aparecen en la figura 308.

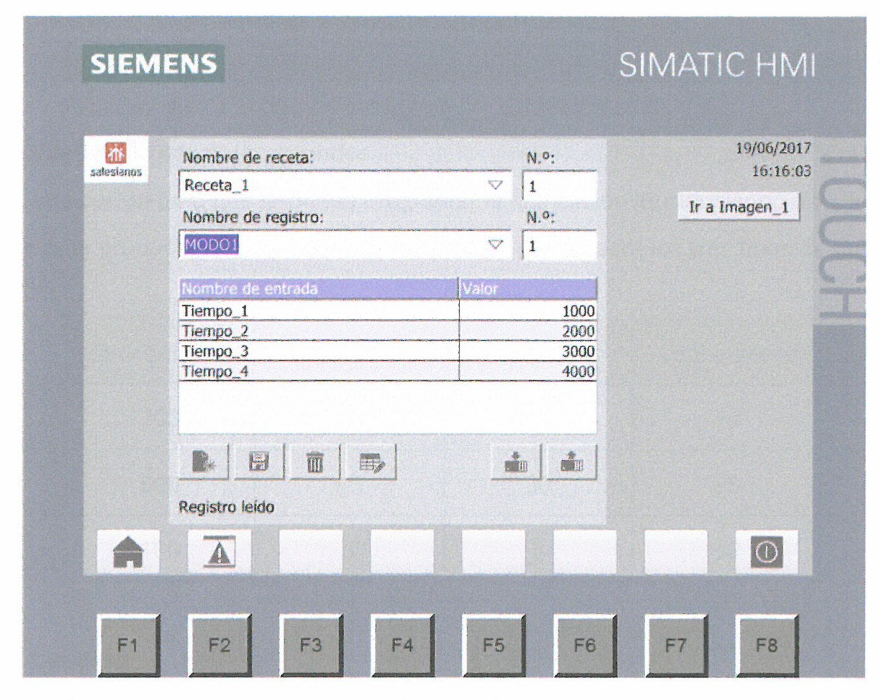


Figura 308

El nombre de la receta aparece solo una vez porque se ha creado una sola receta. En los nombres de los registros, se observan los cuatro modos que se podrán seleccionar. Una vez escogido un modo, en la tabla se muestran los tiempos correspondientes a dichos modos. Por último, hay que enviarlo al PLC pulsando en el icono .

Si se visualiza el DB y se colocan las gafas, se observará que los tiempos se actualizan cuando se envía desde la pantalla. Iniciando con la entrada de *Arranque*, las salidas permanecerán en marcha el tiempo preestablecido a dicho modo.

Visualización de más de una receta en el mismo visor

Para que se visualice más de una receta en el mismo visor, se debe asegurar de que el campo *Receta* de la carpeta *General* de las *Propiedades* se halle vacío. Véase la figura 309.

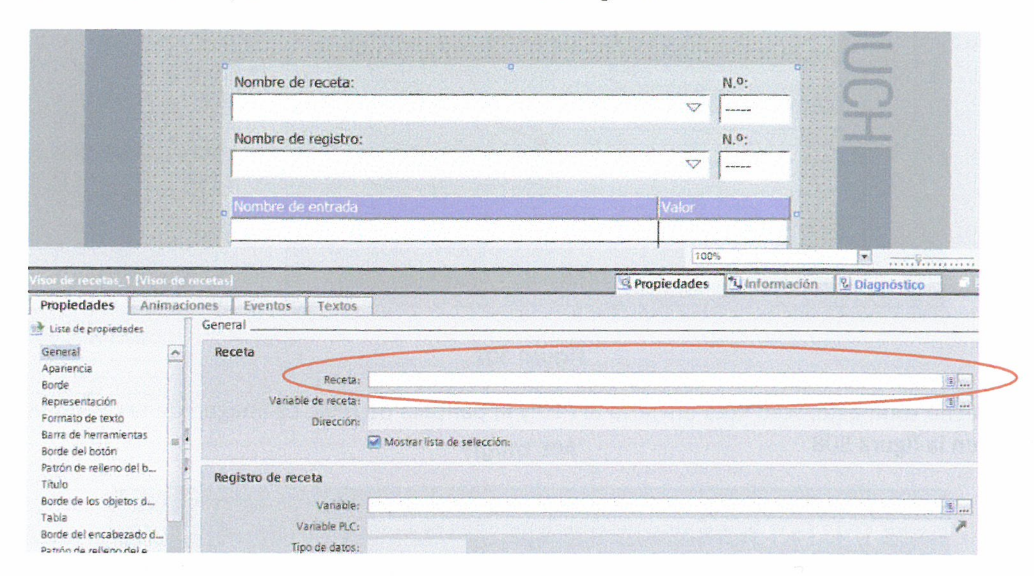


Figura 309

Unidad 6 · Ejercicios prácticos de HMI

22. Fábrica de bebidas

Enunciado

Se trata de realizar un programa y de añadir una pantalla de HMI para producir tres bebidas de sabor naranja, limón y manzana. Cada una de ellas se fabrica en varios formatos, que son zumo, néctar y refresco. Para elaborar cada uno de estos productos se necesitan los siguientes ingredientes: concentrado (de naranja, limón y manzana), azúcar, agua y conservantes.

Una vez que todos los ingredientes han caído al recipiente mezclador, un agitador los mezcla. Las cantidades se miden por tiempo de apertura de las tolvas.

Se deben representar en una pantalla de HMI las tolvas de los productos. Se dispone de una puesta en servicio y de un pulsador de arranque de la producción. Las tolvas se abren con diferentes electroválvulas que actúan sobre los cilindros de apertura.

En la pantalla se han de mostrar los siguientes avisos:

- Aviso de tolva vacía: avisa cuando falta el 10% de la sustancia.
- Alarma de que la tolva no abre o no cierra.

En el caso de las alarmas, la producción se parará con el fin de evitar la pérdida de materia prima.

Un indicador, por su parte, señala el tipo de bebida que se está fabricando.

Los tiempos se pueden introducir en segundos.

Existe también una opción para que se pueda programar la fabricación introduciendo día y hora.

Los tiempos de cada producto son los siguientes (resultan iguales para los tres tipos de bebidas: naranja, limón y manzana):

	Concentrado	Azúcar	Agua	Conservante	Agitador
Néctar	300	10	50	70	50
Zumo	600	45	30	95	80
Refresco	100	3	5	50	30

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una pantalla SIMATIC HMI KTP 700 BASIC 6AV2 con referencia 123-2GB03-0AX0.
3. *Software* WinCC V13 Advanced.
4. Un cable de Profinet para las conexiones.

Realización

En primer lugar, se organizan las entradas y salidas, así como los bloques que se vayan a utilizar.

Este ejercicio se realizará por partes: primero, la principal de puesta en marcha junto con las recetas introduciendo los valores en milisegundos; posteriormente se añadirán los avisos y, por último, se modificará el programa para poder introducir los tiempos en segundos.

Como bloques se utilizarán los siguientes:

- Una función donde se situará el programa de las aperturas y cierres de las tolvas, con sus temporizadores correspondientes.
- En el bloque principal de OB1, se colocará la activación de los avisos y se llamará a la función con la puesta en servicio de la instalación.
- Se creará un DB donde se pondrán los tiempos de apertura de las distintas tolvas y que podrán ser modificados por las recetas desde la pantalla.

Las variables del PLC que se utilizarán se detallan en la figura 310. Hay que tener en cuenta el número de tolvas de materia prima necesarias. Existen productos comunes para las tres bebidas: azúcar, agua y conservantes. Y, luego, habrá tres tolvas que deberán abrirse en función de la receta que se desee fabricar. Estas son: las tolvas de naranja, de limón y de manzana. De estas tolvas, solo se abrirá una en cada receta. Con lo dicho, se necesitarán seis tolvas.

La tolva de cada concentrado (de naranja, de limón o de manzana) se deberá abrir en función de la receta activada. Para ello, la pantalla pasará al PLC una marca donde constará el número de receta seleccionado. Con esta información, se realizará el programa para activar una tolva u otra. Las tolvas comunes se abrirán siempre en las tres recetas.

Tabla de variables estándar

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	Servicio	Bool	%E0.0	☐	☑	☑	
2	Arranque	Bool	%E0.1	☐	☑	☑	
3	S_Azucar_Fallo_Abrir	Bool	%E0.2	☐	☑	☑	Sensores de fallo de apertura y cierre
4	S_Azucar_Fallo_Cerrar	Bool	%E0.3	☐	☑	☑	
5	S_Agua_Fallo_Abrir	Bool	%E0.4	☐	☑	☑	
6	S_Agua_Fallo_Cerrar	Bool	%E0.5	☐	☑	☑	
7	S_Conservante_Fallo_Abrir	Bool	%E0.6	☐	☑	☑	
8	S_Conservante_Fallo_Cerrar	Bool	%E0.7	☐	☑	☑	
9	S_Naranja_Fallo_Abrir	Bool	%E1.0	☐	☑	☑	
10	S_Naranja_Fallo_Cerrar	Bool	%E1.1	☐	☑	☑	
11	S_Limón_Fallo_Abrir	Bool	%E1.2	☐	☑	☑	
12	S_Limón_Fallo_Cerrar	Bool	%E1.3	☐	☑	☑	
13	S_Manzana_Fallo_Abrir	Bool	%E1.4	☐	☑	☑	
14	S_manzana_Fallo_Cerrar	Bool	%E1.5	☐	☑	☑	
15	S_10%_Azucar	Bool	%E1.6	☐	☑	☑	Sensores de 10% para llenarse
16	S_10%_Agua	Bool	%E1.7	☐	☑	☑	
17	S_10%_Conservantes	Bool	%E2.0	☐	☑	☑	
18	S_10%_Naranja	Bool	%E2.1	☐	☑	☑	
19	S_10%_Limón	Bool	%E2.2	☐	☑	☑	
20	S_10%_Manzana	Bool	%E2.3	☐	☑	☑	
21	S_Azucar	Bool	%A0.0	☐	☑	☑	Salidas para las electroválvulas
22	S_Agua	Bool	%A0.1	☐	☑	☑	
23	S_Conservante	Bool	%A0.2	☐	☑	☑	
24	S_Agitador	Bool	%A0.3	☐	☑	☑	
25	S_Naranja	Bool	%A0.4	☐	☑	☑	
26	S_Limón	Bool	%A0.5	☐	☑	☑	
27	S_Manzana	Bool	%A0.6	☐	☑	☑	
28	Activa Agitador	Bool	%M0.0	☐	☑	☑	Diversas marcas para el proceso
29	N_receta	Int	%MW1	☐	☑	☑	
30	Activar_naranja	Bool	%M3.0	☐	☑	☑	
31	Activar_limón	Bool	%M3.1	☐	☑	☑	
32	Activar_manzana	Bool	%M3.2	☐	☑	☑	
33	Memoria_arranque	Bool	%M3.3	☐	☑	☑	
34	Retardo_agitador	Bool	%M3.4	☐	☑	☑	

Figura 310

Las marcas que se utilizan en el programa poseen la misión expuesta a continuación.

Activa Agitador: el agitador se debe poner en marcha al terminar de caer todos los productos si, además se ha pulsado en *Arranque*, una vez puesta en servicio la fabricación. Para evitar que, nada más empezar el

proceso, el agitador arranque (ya que todas las tolvas se hallarán cerradas en ese momento inicial), se ha utilizado un pequeño retardo de 300 ms para que la lógica que arranca el agitador no entre en acción en el primer momento.

La marca *Activa Agitador* arranca el temporizador que mantiene conectado el agitador durante el tiempo necesario.

La lógica que hace que arranque el agitador será: **que no esté abierta** la tolva de naranja o de limón o de manzana (solo una de ellas se encontrará abierta en cada receta) y **que no estén abiertas** las tolvas de azúcar ni la de agua ni la de conservantes. Además, debe activarse también el arranque de la fabricación, que llevará un retraso de 300 ms, con el fin de evitar que se abra en un primer momento, como se ha comentado.

Se debe crear un DB donde se ubiquen todos los tiempos de apertura para cada tolva. El DB creado es el mostrado en la figura 311.

DB_Tiempos (instantánea generada: 23/06/2017 10:42:04)

	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor de a...
1	▼ Static			☐	☐	☐	☐
2	T_Azucar	Time	T#0ms	☐	☑	☑	☐
3	T_Agua	Time	T#0ms	☐	☑	☑	☐
4	T_Conservante	Time	T#0ms	☐	☑	☑	☐
5	T_Agitador	Time	T#0ms	☐	☑	☑	☐
6	T_Naranja	Time	T#0ms	☐	☑	☑	☐
7	T_Limón	Time	T#0ms	☐	☑	☑	☐
8	T_Manzana	Time	T#0ms	☐	☑	☑	☐

Figura 311

La parte del programa que realiza la conexión del agitador al caer todos los productos en la concentradora es la siguiente:

```
// Se setea el arranque para su uso
//en el temporizador TON
        U       "Arranque"                         %E0.1
        S       "Memoria_arranque"                 %M3.3
//Lógica de activación del agitador
//el pequeño retardo de 300 ms.
//También se resetean la memoria de
//arranque.
        UN      "S_Naranja"                        %A0.4
        ON      "S_Limón"                          %A0.5
        ON      "S_Manzana"                        %A0.6
        U       "Retardo_agitador"                 %M3.4
        UN      "S_Azucar"                         %A0.0
        UN      "S_Agua"                           %A0.1
        UN      "S_Conservante"                    %A0.2
        =       "Activa Agitador"                  %M0.0
        R       "Memoria_arranque"                 %M3.3

//Retardo de 300 ms que lo activa el
//arranque de la fabricación
        CALL  TON , "IEC_Timer_0_DB"               %DB3
            Time
            IN :="Memoria_arranque"                %M3.3
            PT :=T#300MS                         T#300MS
            Q  :="Retardo_agitador"                %M3.4
            ET :=
```

N_receta: Constituye una marca de palabra (MW1), de modo que se pasa el número de receta desde la pantalla de HMI.

Activar_naranja, Activar_limón, Activar_manzana: son las marcas que activan la tolva correspondiente al concentrado de cada receta. Depende del número de receta que devuelva la pantalla y, para eso, en el bloque OB1 se realizarán las comparaciones correspondientes.

Memoria_arranque: se corresponde con la marca seteada para mantener el recuerdo de haber arrancado. Resulta necesaria para el temporizador TON.

Retardo_agitador: con esta marca, se pone en marcha el agitador. Este se conecta 300 ms después de haber activado todas las tolvas de producto.

El grupo de programas del PLC realizados hasta ahora son los enumerados a continuación.

OB1:

```
        U      "Servicio"                          %E0.0
        CC     "F_Arranque"                        %FC1

```

FC1:

```
// Se setea el arranque para su uso
//en el temporizador TON
        U      "Arranque"                          %E0.1
        S      "Memoria_arranque"                  %M3.3
//Lógica de activación del agitador
//el pequeño retardo de 300 ms.
//También se resetean la memoria de
//arranque.
        UN     "S_Naranja"                         %A0.4
        ON     "S_Limón"                           %A0.5
        ON     "S_Manzana"                         %A0.6
        U      "Retardo_agitador"                  %M3.4
        UN     "S_Azucar"                          %A0.0
        UN     "S_Agua"                            %A0.1
        UN     "S_Conservante"                     %A0.2
        =      "Activa Agitador"                   %M0.0
        R      "Memoria_arranque"                  %M3.3

//Retardo de 300 ms que lo activa el
//arranque de la fabricación
        CALL  TON , "IEC_Timer_0_DB"               %DB3
           Time
           IN :="Memoria_arranque"                 %M3.3
           PT :=T#300MS                            T#300MS
           Q  :="Retardo_agitador"                 %M3.4
           ET :=

//Temporizadores para cada tolva

        CALL  TP , "IEC_Timer_0_DB_4"              %DB2
           Time
           IN :="Activa Agitador"                  %M0.0
           PT :="DB_Tiempos".T_Agitador
           Q  :="S_Agitador"                       %A0.3
           ET :=


        CALL  TP , "IEC_Timer_0_DB_5"              %DB7
           Time
           IN :="Arranque"                         %E0.1
           PT :="DB_Tiempos".T_Agua
           Q  :="S_Agua"                           %A0.1
           ET :=

        CALL  TP , "IEC_Timer_0_DB_6"              %DB8
           Time
           IN :="Arranque"                         %E0.1
           PT :="DB_Tiempos".T_Azucar
           Q  :="S_Azucar"                         %A0.0
           ET :=
        CALL  TP , "IEC_Timer_0_DB_7"              %DB9
           Time
           IN :="Arranque"                         %E0.1
           PT :="DB_Tiempos".T_Conservant
           Q  :="S_Conservante"                    %A0.2
           ET :=

        CALL  TP , "IEC_Timer_0_DB_8"              %DB10
           Time
           IN :="Activar_naranja"                  %M3.0
           PT :="DB_Tiempos".T_Naranja
           Q  :="S_Naranja"                        %A0.4
           ET :=
        CALL  TP , "IEC_Timer_0_DB_9"              %DB11
           Time
           IN :="Activar_limón"                    %M3.1
           PT :="DB_Tiempos".T_Limón
           Q  :="S_Limón"                          %A0.5
           ET :=
```



```
        CALL  TP , "IEC_Timer_0_DB_10"             %DB12
           Time
           IN :="Activar_manzana"                  %M3.2
           PT :="DB_Tiempos".T_Manzana
           Q  :="S_Manzana"                        %A0.6
           ET :=

// Se necesita conocer qué receta
//se está ejecutando y para ello se
//utiliza la variable "N_receta"
//donde se pasa, desde el HMI, el
//número de receta y con ello se
//realiza el siguiente código.
//Se activa una marca por cada
//producto si se ha pulsado arranque
//y la receta es la correspondiente:
// 1 para naranja, 2 para limón y 3
//para manazana.

        L      "N_receta"                          %MW1
        L      1                                   1
        ==I
        U      "Arranque"                          %E0.1
        =      "Activar_naranja"                   %M3.0

        L      "N_receta"                          %MW1
        L      2                                   2
        ==I
        U      "Arranque"                          %E0.1
        =      "Activar_limón"                     %M3.1

        L      "N_receta"                          %MW1
        L      3                                   3
        ==I
        U      "Arranque"                          %E0.1
        =      "Activar_manzana"                   %M3.2
```

A continuación, deben crearse las imágenes de la pantalla de HMI. En una imagen deberán aparecer las tolvas y el recipiente mezclador. Cada tolva llevará un led para indicar si se encuentra abierta (led verde) o cerrada (led rojo); lo mismo contendrá el agitador. En otra imagen se colocará el visor de recetas. Habrá que añadir un botón para pasar de una pantalla a otra.

Los dibujos de las tolvas y el depósito se buscan entre los diversos elementos que aparecen en el menú de gráficos, en la parte derecha. Los ledes indicadores pueden ser círculos o elipses del menú *Objetos básicos*. A cada uno de estos objetos se le ha de añadir una propiedad de apariencia, la cual será de color rojo cuando el valor de la variable sea 0 y verde cuando sea 1. La variable corresponde a la salida de cada tolva o agitador. En la figura 312 se muestra la apariencia para la salida *Naranja*.

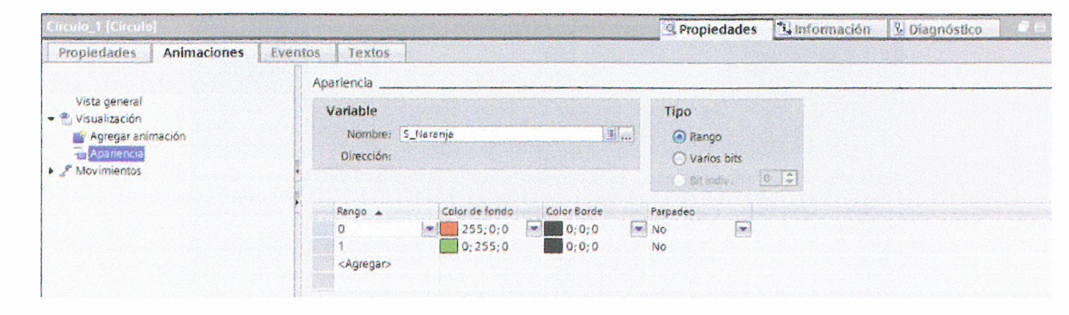


Figura 312

En la figura 313 se representa la imagen de recetas. Se ha modificado el color para darle mayor vistosidad. Esto se hace desde las propiedades del visor de recetas. También se ha modificado el color de fondo de la imagen (se lleva a cabo desde las propiedades de la imagen). Se ha colocado, igualmente, un botón que dirige a la pantalla de inicio.

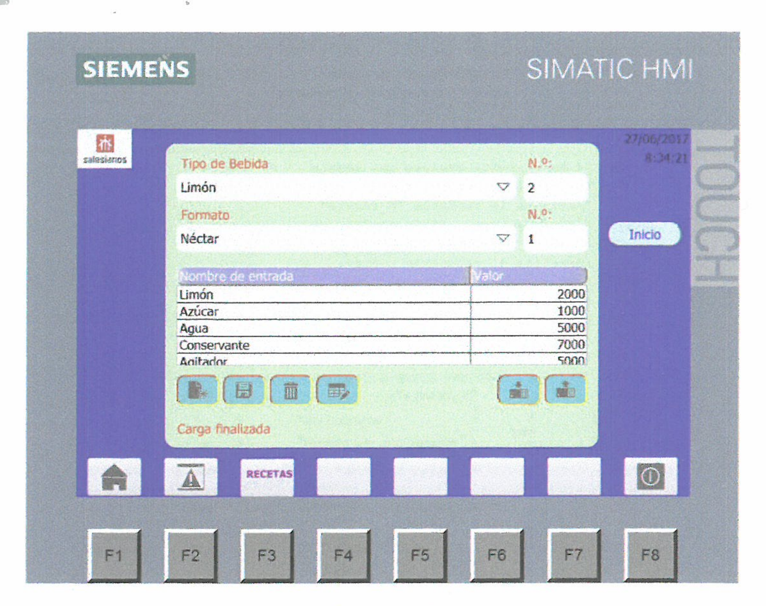


Figura 313

Se debe destacar que a la receta se le ha asignado una variable para introducir el número de la receta y, con ello, se sabrá si ha de abrirse la tolva de naranja, limón o manzana. La variable será de tipo entero, ya que debe guardar un número. La variable será del PLC que previamente se ha definido. En este caso, se corresponde con la variable marca de palabra *N_Receta*. Hay que hacerlo en las propiedades del visor de recetas, en la carpeta general, tal como se indica en la figura 314.

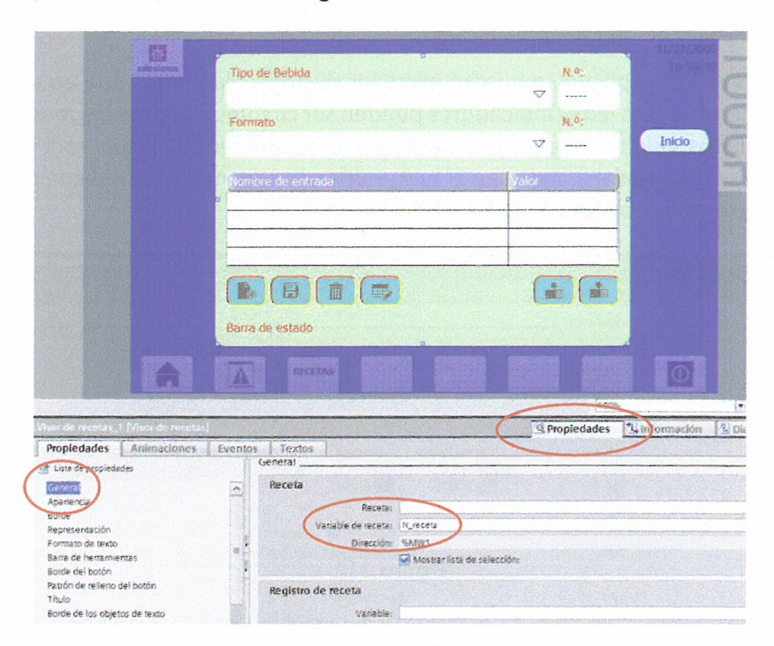


Figura 314

En la figura 315 puede verse la imagen con las tolvas y el agitador. A la tecla F3 se le ha dado la función de dirigirse a la imagen de recetas.

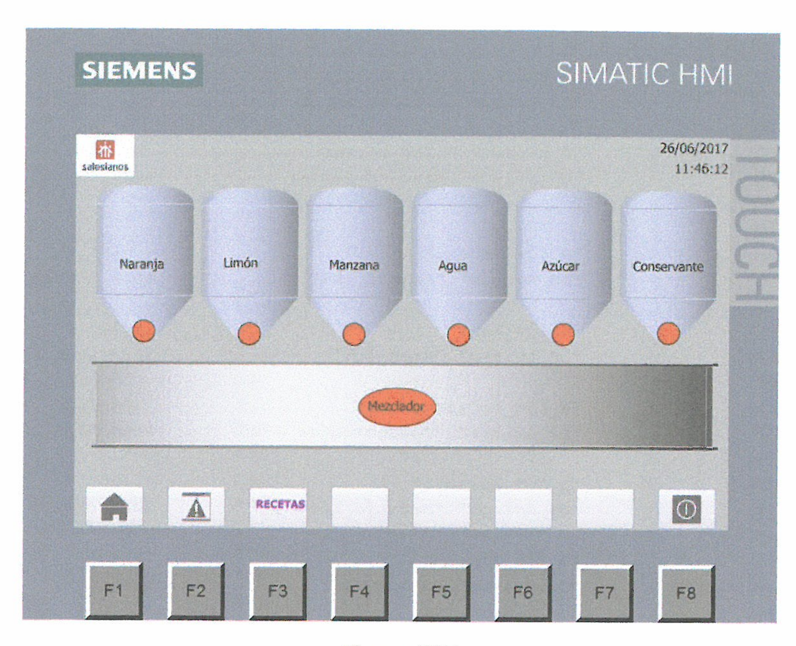


Figura 315

Ya se ha completado la parte central del ejercicio. Ahora se va a proceder a preparar los avisos que se han establecido. Las variables del PLC ya se han creado. Se debe añadir en el OB1 las instrucciones correspondientes a cada uno de los avisos.

El OB1 con las instrucciones que activan las alarmas y avisos queda como se indica a continuación:

```

U     "S_Azucar_Fallo_Abrir"                  %E0.2
=     "DB_Avisos".Azucar_abrir            %DB4.DBX0.4

U     "S_Azucar_Fallo_Cerrar"                 %E0.3
=     "DB_Avisos".Azucar_cerrar           %DB4.DBX0.3

U     "S_Agua_Fallo_Abrir"                    %E0.4
=     "DB_Avisos".Agua_abrir              %DB4.DBX0.1

U     "S_Agua_Fallo_Cerrar"                   %E0.5
=     "DB_Avisos".Agua_cerrar             %DB4.DBX0.0

U     "S_Conservante_Fallo_Abrir"             %E0.6
=     "DB_Avisos".Conservante_abrir       %DB4.DBX0.2

U     "S_Conservante_Fallo_Cerrar"            %E0.7
=     "DB_Avisos".Conservante_cerrar      %DB4.DBX0.5

U     "S_Naranja_Fallo_Abrir"                 %E1.0
=     "DB_Avisos".Naranja_abrir           %DB4.DBX0.6

U     "S_Naranja_Fallo_Cerrar"                %E1.1
=     "DB_Avisos".Naranja_cerrar          %DB4.DBX0.7

U     "S_Limón_Fallo_Abrir"                   %E1.2
=     "DB_Avisos".Limón_abrir             %DB4.DBX1.0

U     "S_Limón_Fallo_Cerrar"                  %E1.3
=     "DB_Avisos".Limón_cerrar            %DB4.DBX1.1
```

```
U     "S_Manzana_Fallo_Abrir"              %E1.4
=     "DB_Avisos".Manzana_abrir            %DB4.DBX1.2

U     "S_manzana_Fallo_Cerrar"             %E1.5
=     "DB_Avisos".Manzana_cerrar           %DB4.DBX1.3


U     "S_10%_Azucar"                       %E1.6
=     "DB_Avisos"."Azucar_10%"             %DB4.DBX1.4

U     "S_10%_Agua"                         %E1.7
=     "DB_Avisos"."Agua_10%"               %DB4.DBX1.5

U     "S_10%_Conservantes"                 %E2.0
=     "DB_Avisos"."Conservantes_10%"       %DB4.DBX1.6

U     "S_10%_Naranja"                      %E2.1
=     "DB_Avisos"."Naranja_10%"            %DB4.DBX1.7

U     "S_10%_Limón"                        %E2.2
=     "DB_Avisos"."Limón_10%"              %DB4.DBX2.0

U     "S_10%_Manzana"                      %E2.3
=     "DB_Avisos"."Manzana_10%"            %DB4.DBX2.1


U     "Servicio"                           %E0.0
CC    "F_Arranque"                         %FC1
```

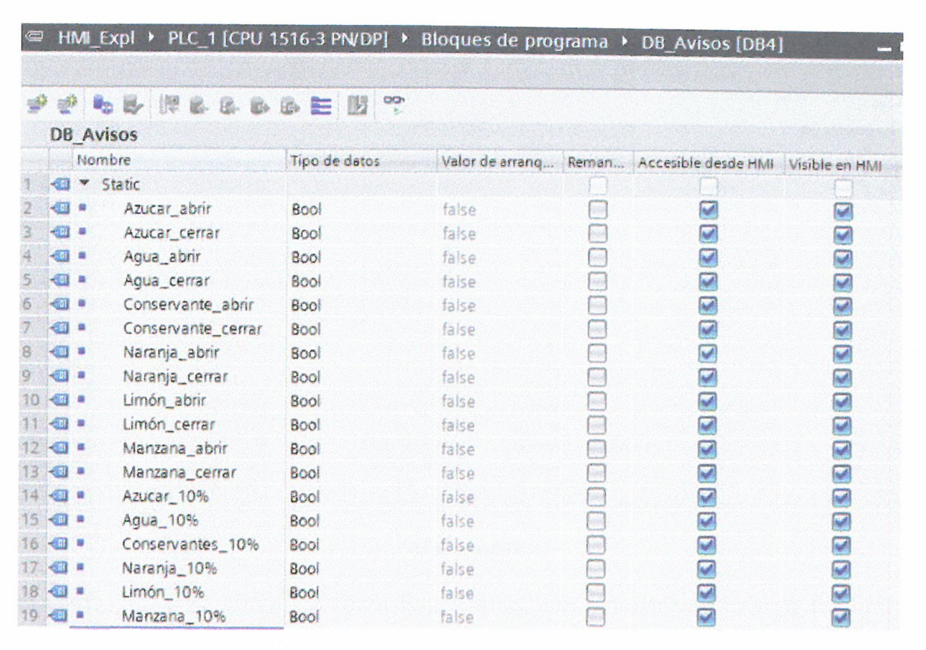
Los fallos de apertura y cierre de las electroválvulas se ponen como alarmas (con recuse) y los avisos de falta del 10 % para llenarse serán *warnings*.

El DB de los avisos y alarmas se refleja en la figura 316. No hay que olvidarse de cambiar las propiedades del DB de avisos; para ello, se desmarca la pestaña *Acceso optimizado al bloque*. Existen 18 avisos, por lo que se necesitarán dos variables de palabra (Word). Se debe recordar que el *byte* de menor peso estará colocado en el *byte* de mayor peso del conjunto. En la siguiente tabla se puede comprobar la distribución de bits dentro del conjunto de cada una de las variables de palabra.

	***Byte* 0**								***Byte* 1**							
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Las variables que deben definirse serán para este caso la DB4.DBW0 y la DB4.DBW2. Según se vio en la parte de explicación de la configuración de los avisos, hay que indicar el bit que ocupa cada aviso dentro del DB.

En este caso, el primer aviso, que es *Azucar_abrir*, ocupa el bit 8 de la variable DB4.DBW0, y el aviso *Naranja_cerrar* será el bit 15. El aviso *Limón*_abrir será el bit 0 de esa misma variable, DB4.DBW0. El aviso *Limón_10%* será el bit 0 de la palabra DB4.DBW2. Y así con todos los demás avisos.



	Nombre	Tipo de datos	Valor de arranq...	Reman...	Accesible desde HMI	Visible en HMI
1	▼ Static					
2	Azucar_abrir	Bool	false	☐	☑	☑
3	Azucar_cerrar	Bool	false	☐	☑	☑
4	Agua_abrir	Bool	false	☐	☑	☑
5	Agua_cerrar	Bool	false	☐	☑	☑
6	Conservante_abrir	Bool	false	☐	☑	☑
7	Conservante_cerrar	Bool	false	☐	☑	☑
8	Naranja_abrir	Bool	false	☐	☑	☑
9	Naranja_cerrar	Bool	false	☐	☑	☑
10	Limón_abrir	Bool	false	☐	☑	☑
11	Limón_cerrar	Bool	false	☐	☑	☑
12	Manzana_abrir	Bool	false	☐	☑	☑
13	Manzana_cerrar	Bool	false	☐	☑	☑
14	Azucar_10%	Bool	false	☐	☑	☑
15	Agua_10%	Bool	false	☐	☑	☑
16	Conservantes_10%	Bool	false	☐	☑	☑
17	Naranja_10%	Bool	false	☐	☑	☑
18	Limón_10%	Bool	false	☐	☑	☑
19	Manzana_10%	Bool	false	☐	☑	☑

Figura 316

Se deberán crear las dos variables de HMI indicadas anteriormente: DB4.DBW0 y DB4.DBW2. Lógicamente, esto dependerá de la configuración de cada uno. Hay que asegurarse de que la variable creada corresponde al DB creado (DB4, en este caso) y a la palabra 0 y 2.

En la figura 317 se indica el proceso para crear las variables del tipo HMI. En dicha figura solo aparece el proceso para una variable. Se deberá proceder de la misma manera con la segunda variable.

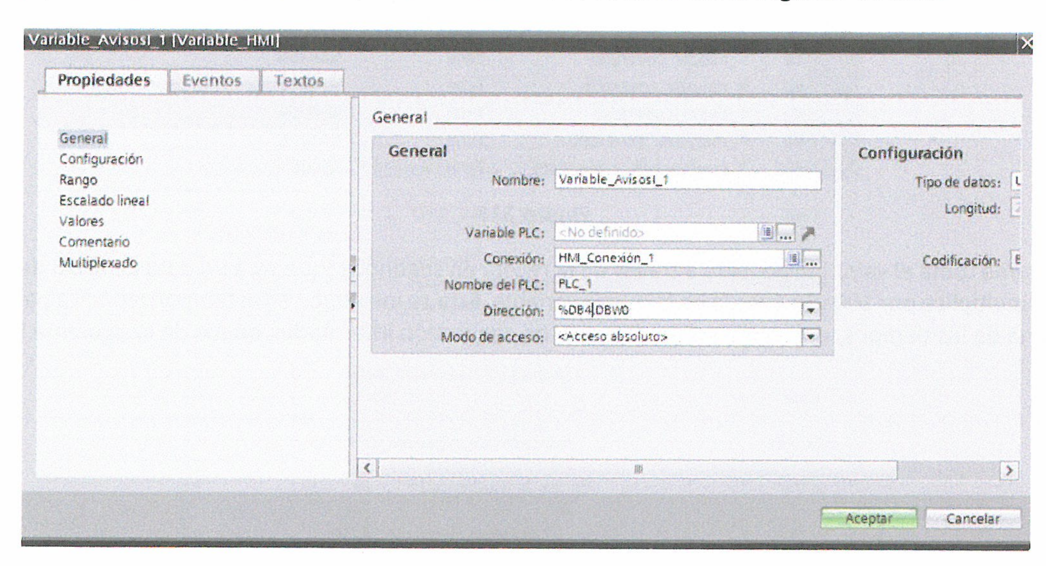
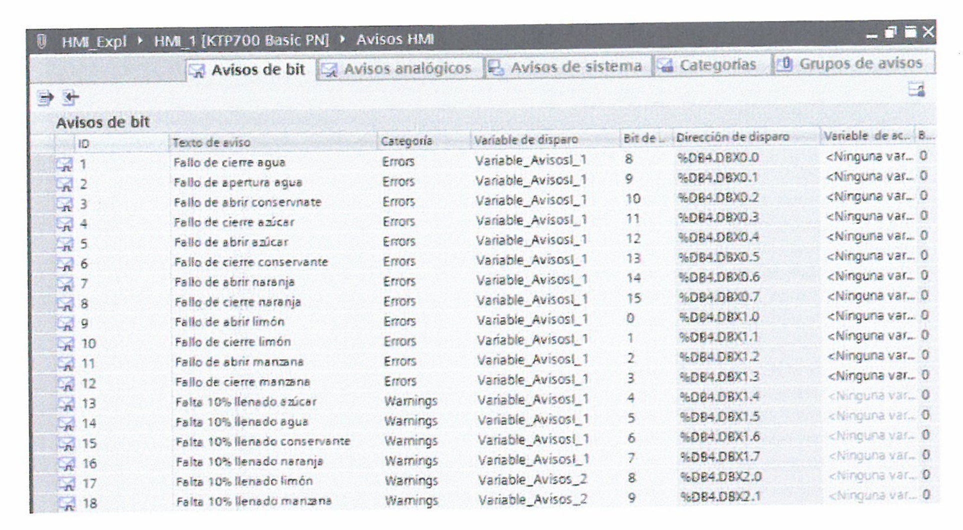


Figura 317

Los avisos quedarán como se indica en la figura 318.



HMI_Expl ▸ HMI_1 [KTP700 Basic PN] ▸ Avisos HMI

Avisos de bit | Avisos analógicos | Avisos de sistema | Categorías | Grupos de avisos

Avisos de bit

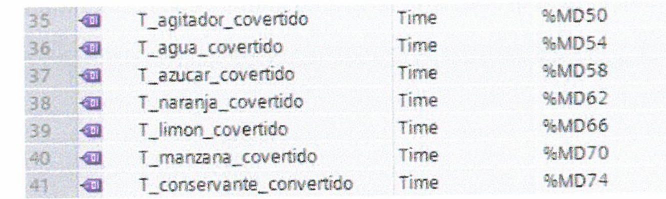
ID	Texto de aviso	Categoría	Variable de disparo	Bit de ...	Dirección de disparo	Variable de ac...	B...
1	Fallo de cierre agua	Errors	Variable_AvisosI_1	8	%DB4.DBX0.0	<Ninguna var...	0
2	Fallo de apertura agua	Errors	Variable_AvisosI_1	9	%DB4.DBX0.1	<Ninguna var...	0
3	Fallo de abrir conservnate	Errors	Variable_AvisosI_1	10	%DB4.DBX0.2	<Ninguna var...	0
4	Fallo de cierre azúcar	Errors	Variable_AvisosI_1	11	%DB4.DBX0.3	<Ninguna var...	0
5	Fallo de abrir azúcar	Errors	Variable_AvisosI_1	12	%DB4.DBX0.4	<Ninguna var...	0
6	Fallo de cierre conservante	Errors	Variable_AvisosI_1	13	%DB4.DBX0.5	<Ninguna var...	0
7	Fallo de abrir naranja	Errors	Variable_AvisosI_1	14	%DB4.DBX0.6	<Ninguna var...	0
8	Fallo de cierre naranja	Errors	Variable_AvisosI_1	15	%DB4.DBX0.7	<Ninguna var...	0
9	Fallo de abrir limón	Errors	Variable_AvisosI_1	0	%DB4.DBX1.0	<Ninguna var...	0
10	Fallo de cierre limón	Errors	Variable_AvisosI_1	1	%DB4.DBX1.1	<Ninguna var...	0
11	Fallo de abrir manzana	Errors	Variable_AvisosI_1	2	%DB4.DBX1.2	<Ninguna var...	0
12	Fallo de cierre manzana	Errors	Variable_AvisosI_1	3	%DB4.DBX1.3	<Ninguna var...	0
13	Falta 10% llenado azúcar	Warnings	Variable_AvisosI_1	4	%DB4.DBX1.4	<Ninguna var...	0
14	Falta 10% llenado agua	Warnings	Variable_AvisosI_1	5	%DB4.DBX1.5	<Ninguna var...	0
15	Falta 10% llenado conservante	Warnings	Variable_AvisosI_1	6	%DB4.DBX1.6	<Ninguna var...	0
16	Falta 10% llenado naranja	Warnings	Variable_AvisosI_1	7	%DB4.DBX1.7	<Ninguna var...	0
17	Falta 10% llenado limón	Warnings	Variable_Avisos_2	8	%DB4.DBX2.0	<Ninguna var...	0
18	Falta 10% llenado manzana	Warnings	Variable_Avisos_2	9	%DB4.DBX2.1	<Ninguna var...	0

Figura 318

Si se utiliza la *Imagen general*, se recuerda que, para visualizar los *warnings,* se ha de activar esa opción en las propiedades de los visores de avisos en la *Imagen general*.

Modificación para entrar los tiempos en segundos

Se va a modificar el programa de la función de arranque (FC1) para que se puedan introducir en la receta de la pantalla los valores en segundos en vez de milisegundos. Para ello, se crearán siete variables más de tiempo. Serán siete marcas de doble palabra con el tipo de dato *time*. En la figura 319 se muestran esas nuevas variables.



35	T_agitador_covertido	Time	%MD50
36	T_agua_covertido	Time	%MD54
37	T_azucar_covertido	Time	%MD58
38	T_naranja_covertido	Time	%MD62
39	T_limon_covertido	Time	%MD66
40	T_manzana_covertido	Time	%MD70
41	T_conservante_convertido	Time	%MD74

Figura 319

El programa toma el valor introducido a través de la receta en segundos (que será un dato de la DB de tiempos), lo multiplica por 100 y lo guarda en la nueva variable. Esta se introduce en el temporizador. La parte del programa de los tiempos, una vez añadido ese paso de conversión intermedio, queda de la siguiente forma:


```
//Temporizadores para cada tolva con conversión
// a milisegundos

      L      "DB_Tiempos".T_Agitador
      L      1000                                   1000
      *D
      T      "T_agitador_covertido"                 %MD50

      CALL  TP , "IEC_Timer_0_DB_4"                 %DB2
         Time
         IN :="Activa Agitador"                     %M0.0
         PT :="T_agitador_covertido"                %MD50
         Q  :="S_Agitador"                          %A0.3
         ET :=

      L      "DB_Tiempos".T_Agua
      L      1000                                   1000
      *D
      T      "T_agua_covertido"                     %MD54

      CALL  TP , "IEC_Timer_0_DB_5"                 %DB7
         Time
         IN :="Arranque"                            %E0.1
         PT :="T_agua_covertido"                    %MD54
         Q  :="S_Agua"                              %A0.1
         ET :=

      L      "DB_Tiempos".T_Azucar
      L      1000                                   1000
      *D
      T      "T_azucar_covertido"                   %MD58

      CALL  TP , "IEC_Timer_0_DB_6"                 %DB8
         Time
         IN :="Arranque"                            %E0.1
         PT :="T_azucar_covertido"                  %MD58
         Q  :="S_Azucar"                            %A0.0
         ET :=
```

```

      L      "DB_Tiempos".T_Conservante
      L      1000                                   1000
      *D
      T      "T_conservante_convertido"             %MD74

      CALL  TP , "IEC_Timer_0_DB_7"                 %DB9
         Time
         IN :="Arranque"                            %E0.1
         PT :="T_conservante_convertido"            %MD74
         Q  :="S_Conservante"                       %A0.2
         ET :=

      L      "DB_Tiempos".T_Naranja
      L      1000                                   1000
      *D
      T      "T_naranja_covertido"                  %MD62

      CALL  TP , "IEC_Timer_0_DB_8"                 %DB10
         Time
         IN :="Activar_naranja"                     %M3.0
         PT :="T_naranja_covertido"                 %MD62
         Q  :="S_Naranja"                           %A0.4
         ET :=

      L      "DB_Tiempos".T_Limón
      L      1000                                   1000
      *D
      T      "T_limon_covertido"                    %MD66

      CALL  TP , "IEC_Timer_0_DB_9"                 %DB11
         Time
         IN :="Activar_limón"                       %M3.1
         PT :="T_limon_covertido"                   %MD66
         Q  :="S_Limón"                             %A0.5
         ET :=
```

```
L     "DB_Tiempos".T_Manzana
L     1000                                      1000
*D
T     "T_manzana_covertido"                     %MD70

CALL  TP , "IEC_Timer_0_DB_10"                  %DB12
   Time
   IN :="Activar_manzana"                       %M3.2
   PT :="T_manzana_covertido"                   %MD70
   Q  :="S_Manzana"                             %A0.6
   ET :=
```

Todos los valores de tiempo en milisegundos introducidos en las recetas se deben cambiar a segundos.

No hay que olvidar cargarlo todo en PLC y HMI.

Unidad 7 Ejercicios prácticos de industria 4.0



En esta unidad veremos:

23. Red industrial inalámbrica (WLAN)
24. Envío de *e-mails* desde HMI (WinCC)
25. Envío de SMS (SINAUT)
26. IoT2040 Configuración
27. IoT2040 Conexión con PLC. Lectura
28. IoT2040 Conexión con PLC. Escritura
29. IoT2040 *e-mails* y Twitter

```
    L       "DB_Tiempos".T_Manzana
    L       1000                                    1000
    *D
    T       "T_manzana_covertido"                   %MD70

    CALL  TP , "IEC_Timer_0_DB_10"                  %DB12
        Time
        IN  :="Activar_manzana"                     %M3.2
        PT  :="T_manzana_covertido"                 %MD70
        Q   :="S_Manzana"                           %A0.6
        ET  :=
```

Todos los valores de tiempo en milisegundos introducidos en las recetas se deben cambiar a segundos.

No hay que olvidar cargarlo todo en PLC y HMI.

Unidad 7 Ejercicios prácticos de industria 4.0



En esta unidad veremos:

23. Red industrial inalámbrica (WLAN)

Introducción

El beneficio de utilizar redes inalámbricas en la industria resulta evidente. La eliminación de cables en aquellas instalaciones donde el cableado siempre ha sido un problema conlleva instalaciones sencillas de instalar y de mantener. Si se piensa en una grúa de pluma, enseguida se puede comprobar lo importante de utilizar una red inalámbrica. En general, en todas las instalaciones donde hay movimiento y los cables son arrastrados, el uso de este tipo de redes cuenta con una implementación perfecta.

Naturalmente, también existen muchos problemas añadidos, como las interferencias y la seguridad. La seguridad tiene un problema, ya que estas redes inalámbricas suelen conectarse también a Internet. Pero tales problemas se pueden minimizar si se realiza una instalación implementando todas las medidas de seguridad en redes y en el cálculo correcto de su alcance.

Para realizar este ejercicio es necesario la utilización de un Punto de acceso y de un Cliente, los dos de tipo industrial.

Enunciado

Se va a realizar la comunicación entre un PLC y una ET vía *wireless*. Primero debe realizarse la instalación con cable y, posteriormente, conectar el Punto de acceso y el Cliente.

Material necesario

1. Una CPU 1516-3PN/DP con los siguientes módulos centralizados:
   - Una fuente de alimentación PM de 190 W.
   - Un módulo de 32 entradas digitales a 24 V (DC).
   - Un módulo de 32 salidas digitales a 24 V (DC).
   - Un módulo de 8 entradas analógicas.
   - Un módulo de 4 salidas analógicas.
2. Una estación de trabajo descentralizada ET 200S de Siemens y que lleva los siguientes módulos:
   - Cinco módulos de salidas digitales de 4 bits.
   - Cinco módulos de entradas digitales de 4 bits.
3. Punto de acceso SIEMENS.
4. Cliente *wireless* SIEMENS.
5. Un cable de Profinet para las conexiones.

Realización

Lo primero que debe hacerse es comprobar que se disponen de las aplicaciones e interfaz adecuadas en las propiedades de la tarjeta de red del ordenador. En la figura 320 se indica lo que se debe tener instalado.

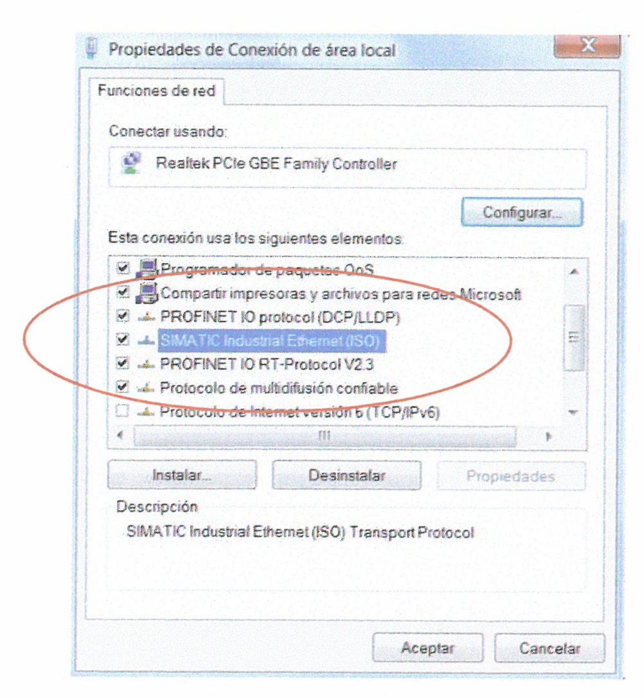


Figura 320

Ahora se debe realizar el montaje y la configuración de un PLC y una ET 200 de forma cableada y comprobar que funciona correctamente.

Una vez funcionando con cable y configurados los Puntos de acceso y Cliente, como se indica posteriormente, se conecta el PLC al punto de acceso y la ET al Cliente. Al poco tiempo de conectar, el PLC y la ET 200 han de funcionar correctamente ya que el Punto de acceso y el Cliente resultan transparentes (figura 321).

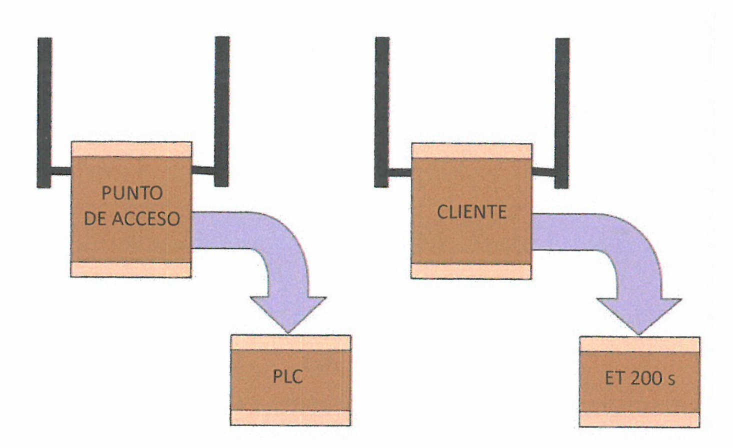


Figura 321

El aspecto de los ledes del Punto de acceso y Cliente deben aparecer de la siguiente forma (para estos dispositivos):

Sin conectar la ET 200 ni el PLC, cuando se haya establecido conexión entre el punto de acceso y el Cliente, los ledes R1 y L2 deberán mostrarse en verde fijo, mientras que los demás se hallarán apagados.

Una vez conectado la ET y el PLC, los ledes P1 y R1 de los dos (Punto de acceso y Cliente) deberán estar en naranja (amarillo), L2 en verde fijo y L1 y F apagados (figura 322).

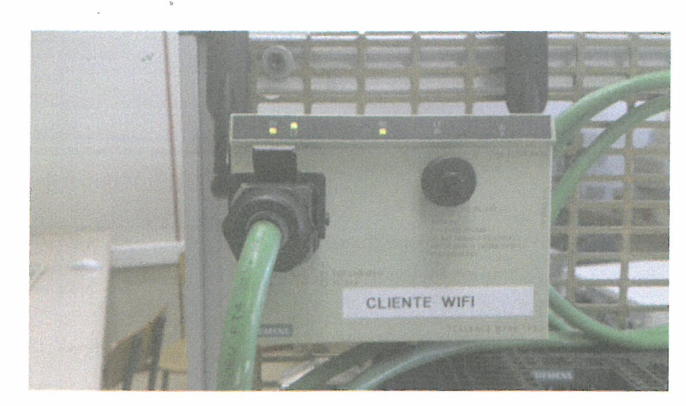


Figura 322

Para asegurarse de si existe conexión wifi con la ET, se puede conectar el Punto de acceso al PC y realizar un PING a la dirección de la ET; si permite el envío y el recibo, significa que se accede. Si el funcionamiento resulta intermitente o no funciona nada, se debe comprobar que el ciclo I/O (propiedades) está puesto en un tiempo de actualización de 4.000, 8.000 o 16.000 ms; con ello, se sincronizan los tiempos de la red con los del acceso a las E/S. Esto se hace tocando en la ET 200S en la línea de la cabecera, luego en *Propiedades*, *General* y en *Opciones avanzadas* (figura 323).

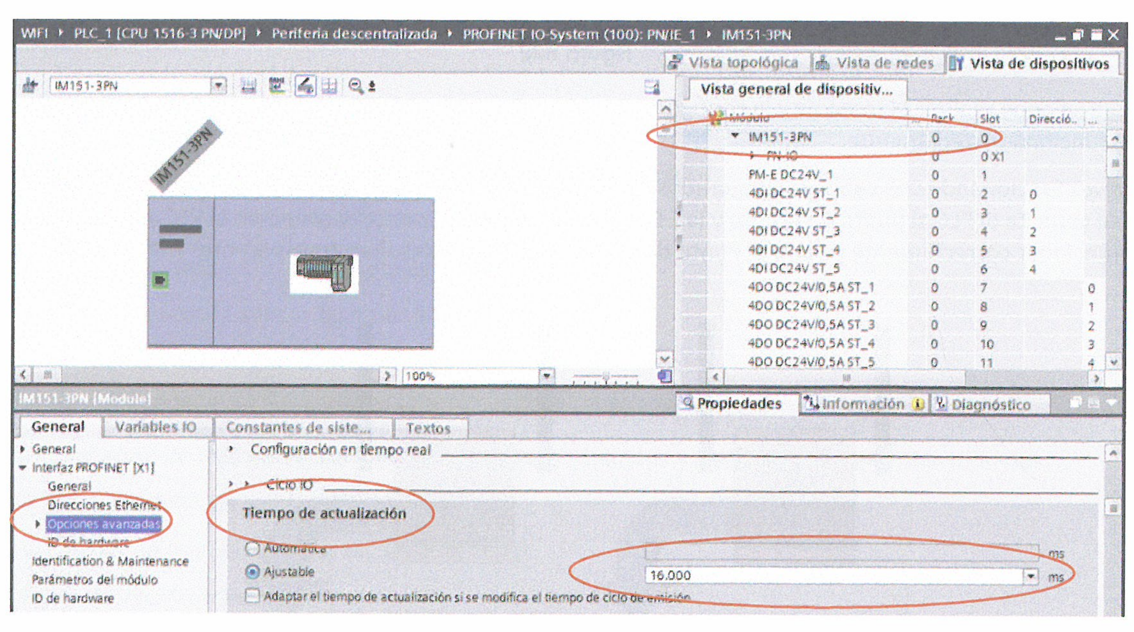


Figura 323

Como se ha dicho, el PLC se conectará al Punto de acceso y la ET 200S se sitúa en el Cliente. Se necesita que el Cliente presente la misma MAC que la ET 200S, lo cual se realiza manualmente desde la web. No es que se cambie la MAC del Cliente, ya que eso no resulta posible; lo que se añade es una MAC adoptada que debe coincidir con la de la ET. Si se comprueba la MAC del Cliente (una vez modificada manualmente), aparece como no cambiada porque lo que se ve es su MAC real y no la adoptada.

Configuración del Punto de acceso y Cliente

Para poder acceder al Punto de acceso o Cliente, debe conectarse con cable a ellos desde el PC y, para ello, es necesario disponer de una dirección dentro del mismo grupo que la tarjeta de red. Se puede utilizar la herramienta PRONETA para cambiar nombres y direcciones o las herramientas del propio TIA PORTAL.

La clave para entrar al Punto de acceso y Cliente, por defecto, es «admin» (en minúsculas), tanto para usuario como para *password*.

Punto de acceso

Una vez que se ha entrado en el Punto de acceso, debe utilizarse el WIZARD BASIC y se configura lo fundamental. En las figuras 324- 326 se representan las ventanas que aparecen en el WIZARD BASIC.

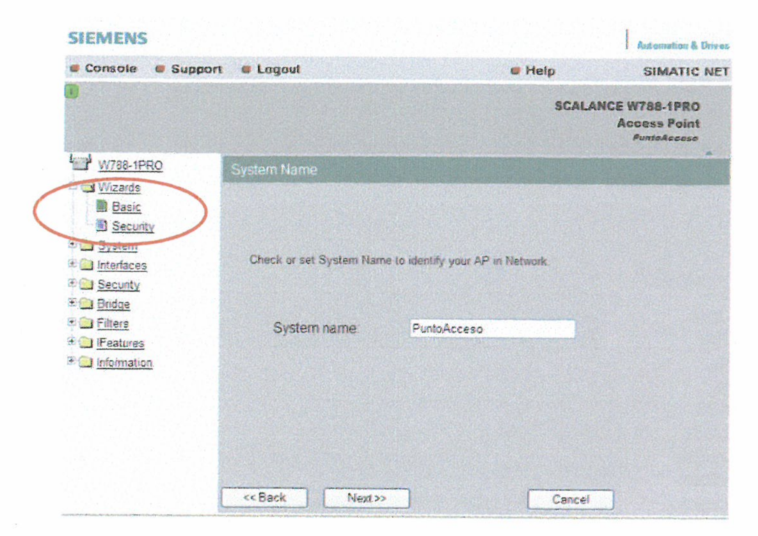


Figura 324

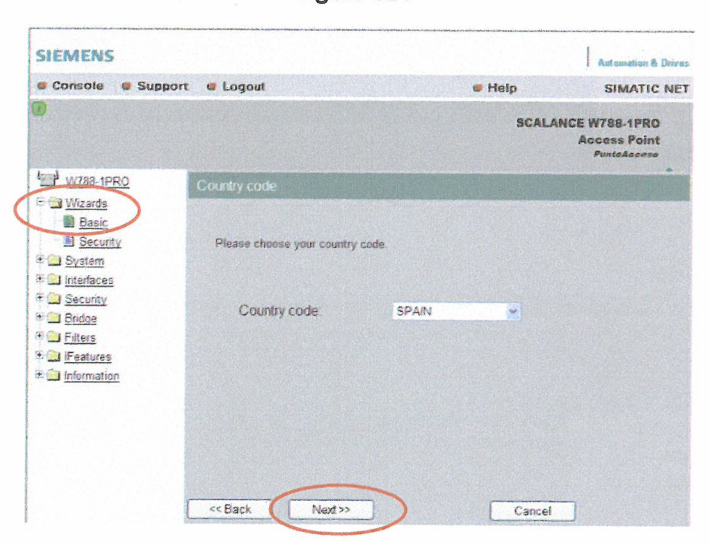


Figura 325

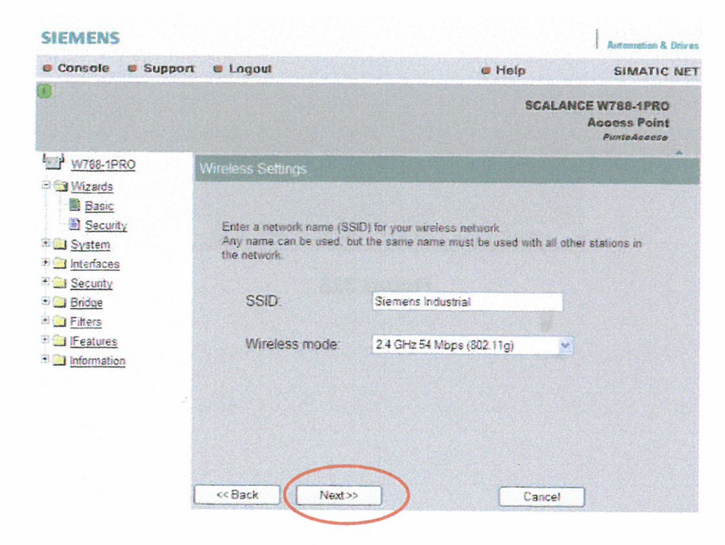


Figura 326

En la figura 327 se puede cambiar el canal de emisión, ya que podría producirse solapamiento (*overlap*) de canales y provocar interferencias. Habría que saber qué canales son los más utilizados en la zona para elegir un canal poco saturado. El Cliente se conectará al canal que se ponga aquí.

Con el programa (*app*) InSSaider, o cualquier otro similar, se observarán los canales colindantes utilizados. También se observa desde el *Punto de acceso* en el menú *Long Table* (en la carpeta *Information*). En las figuras 328-329 se muestra esta opción.

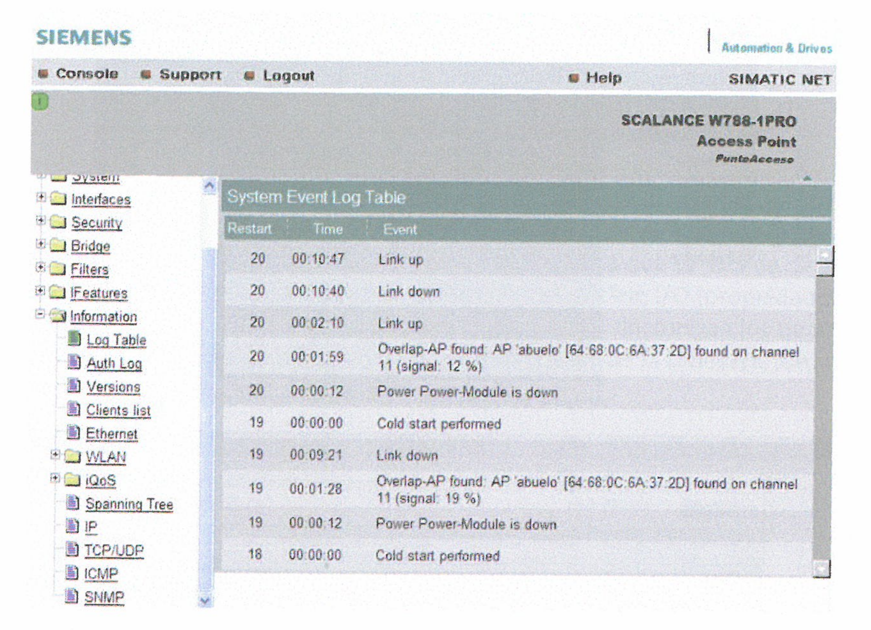


Figura 327

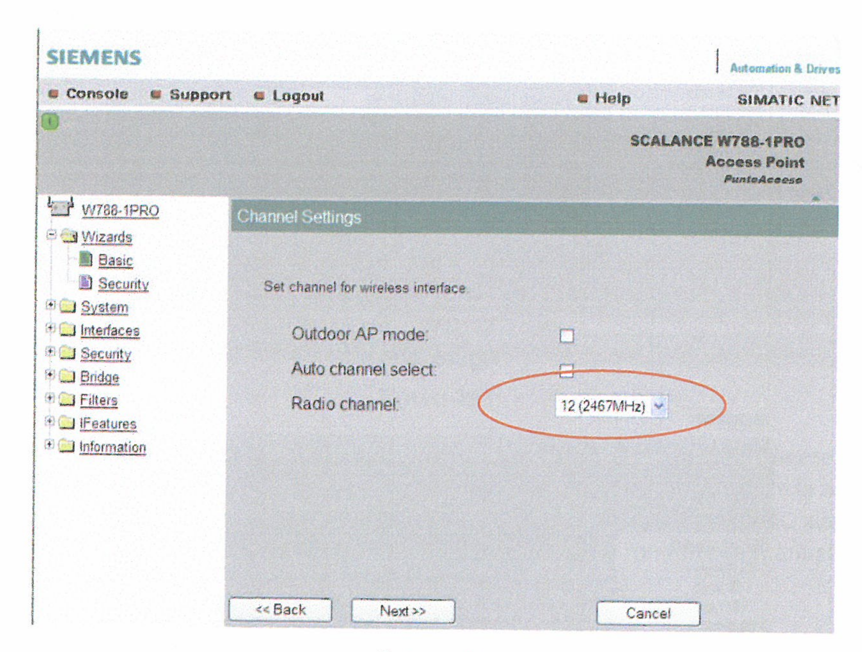


Figura 328

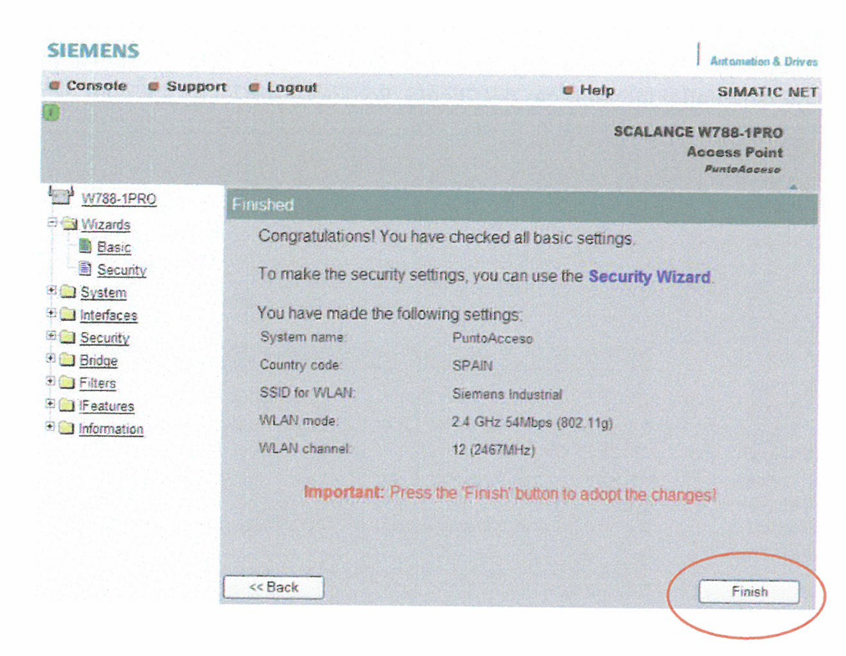


Figura 329

En la figura 330 se puede guardar la configuración del *Punto de acceso*. Posteriormente se podrá cargar cualquier configuración guardada.

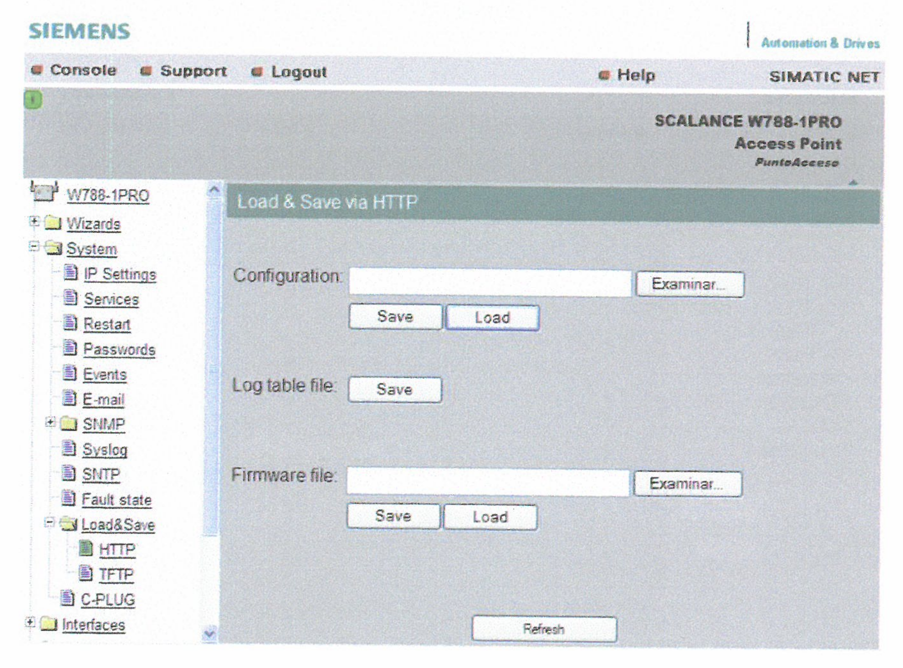


Figura 330

Cliente

Igual que con el *Punto de acceso*, podemos seguir la configuración con el WIZARD BASIC. En las figuras 331-333 se indica cómo deben quedar las ventanas del Cliente. Por supuesto, las direcciones y los nombres serán los propios de cada caso.

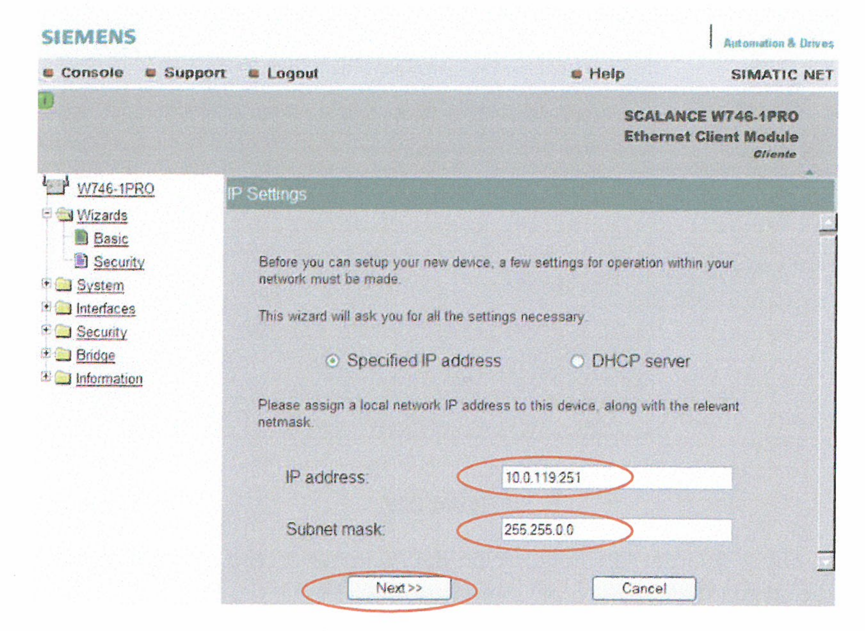


Figura 331

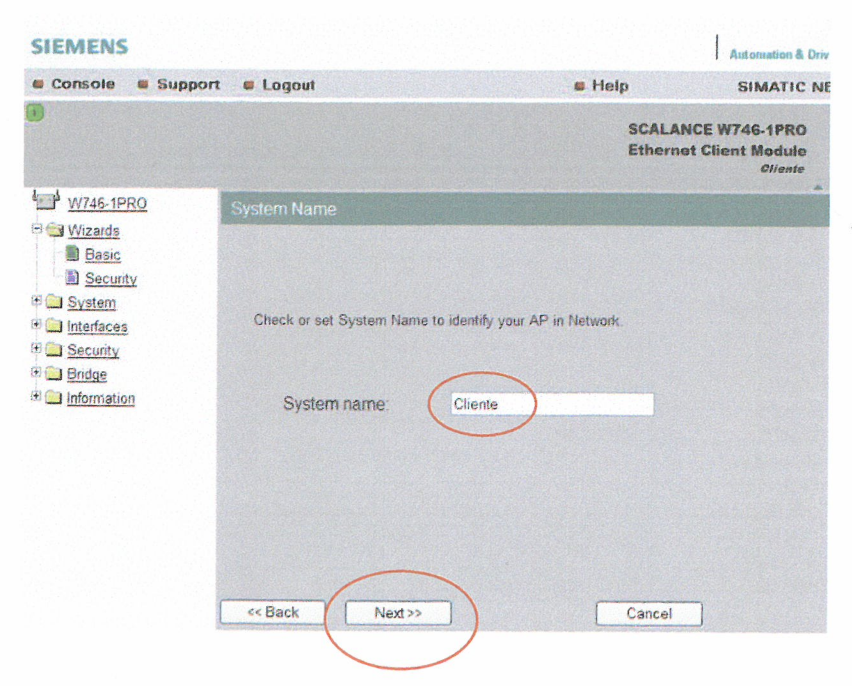


Figura 332

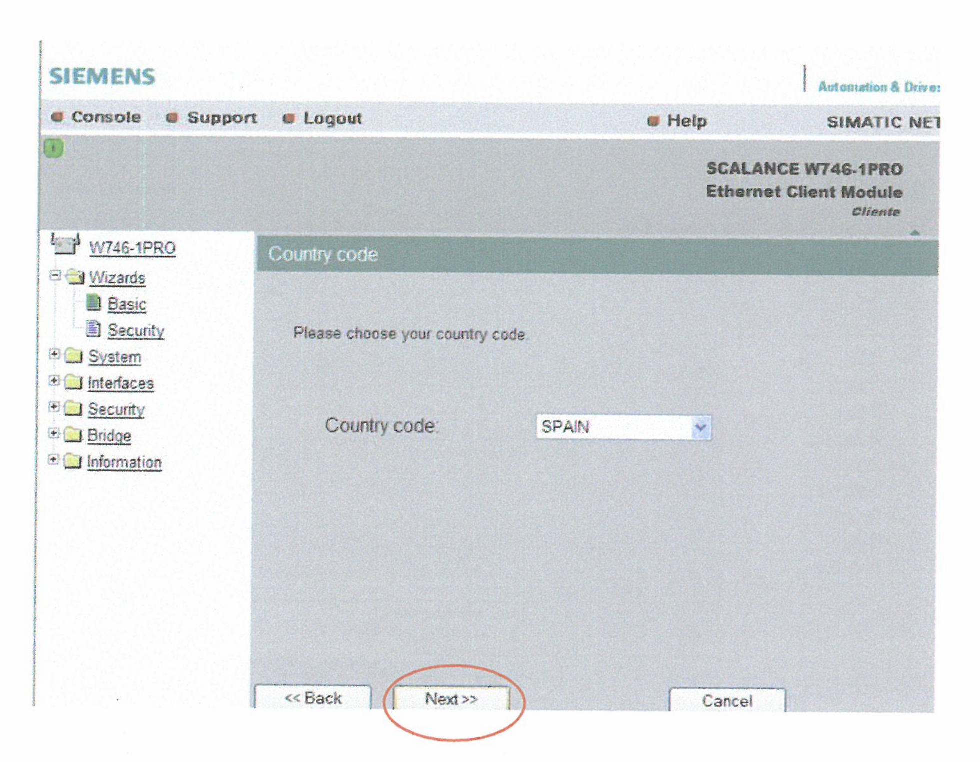


Figura 333

Como se muestra en la figura 333 hay que indicar el nombre de la red con la que debe conectarse; será el identificador de la red (SSID).

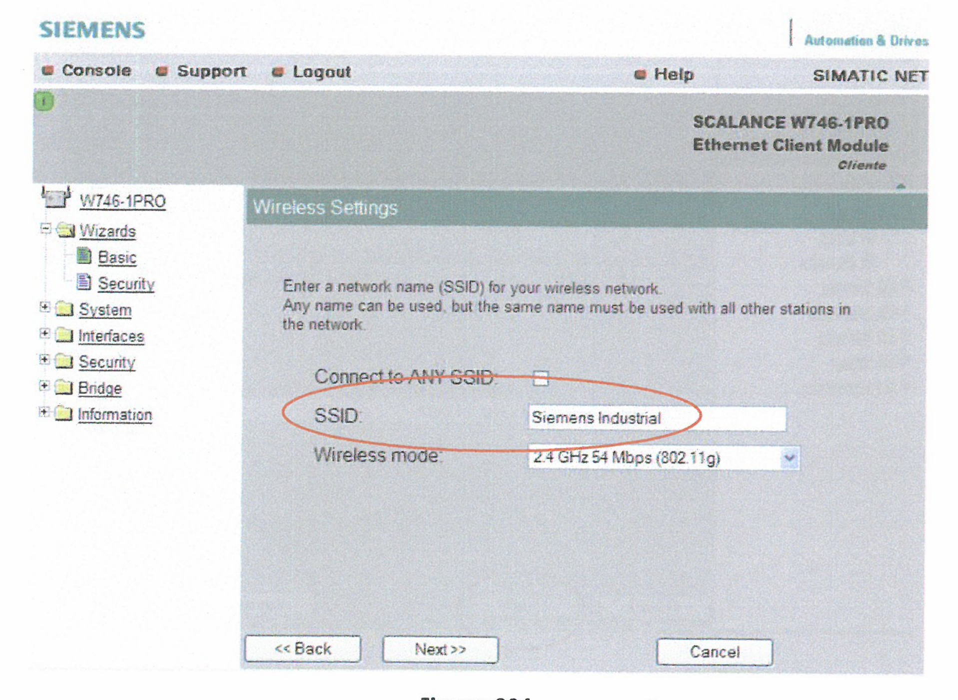


Figura 334

El Cliente debe *adoptar* la MAC de la ET que le ponemos. Si se elige el modo *Layer 2 Tunnel,* se puede situar «aguas abajo» hasta 8 ET. De esta manera, la adopción de la MAC resulta automática (figuras 335-337).

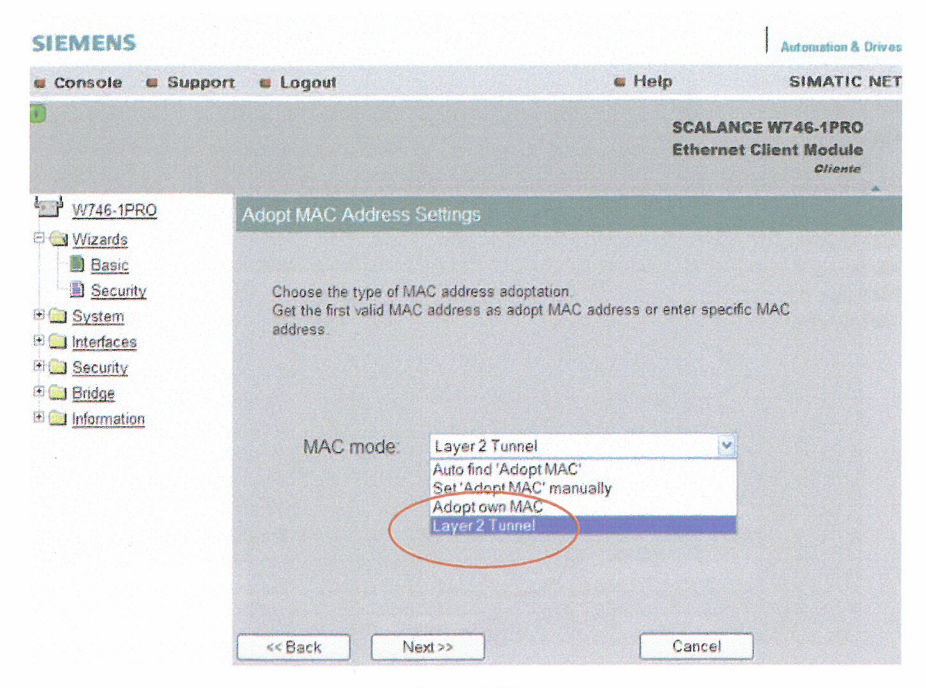


Figura 335

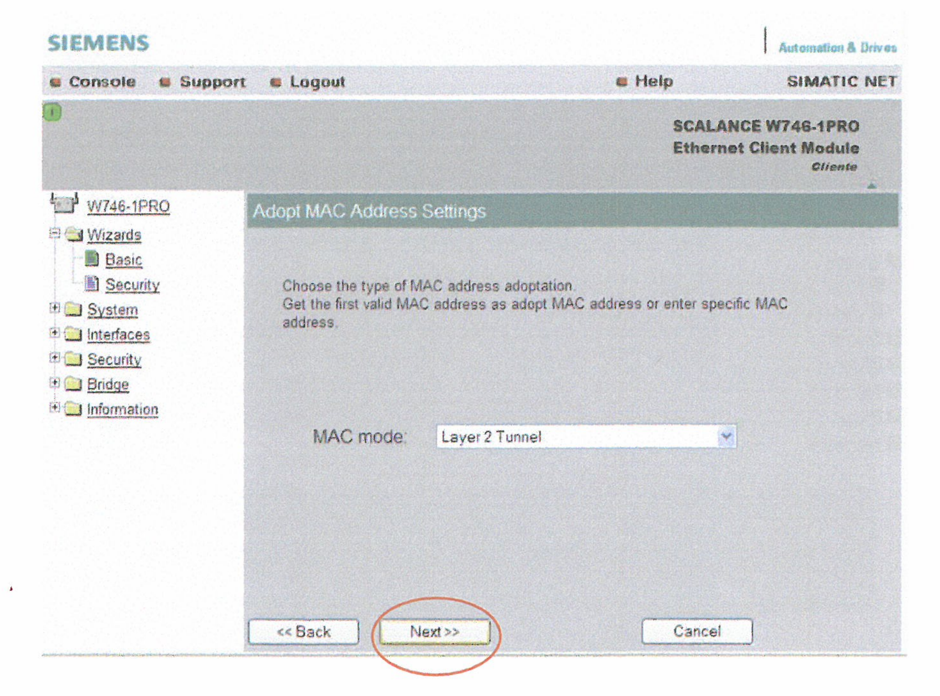


Figura 336

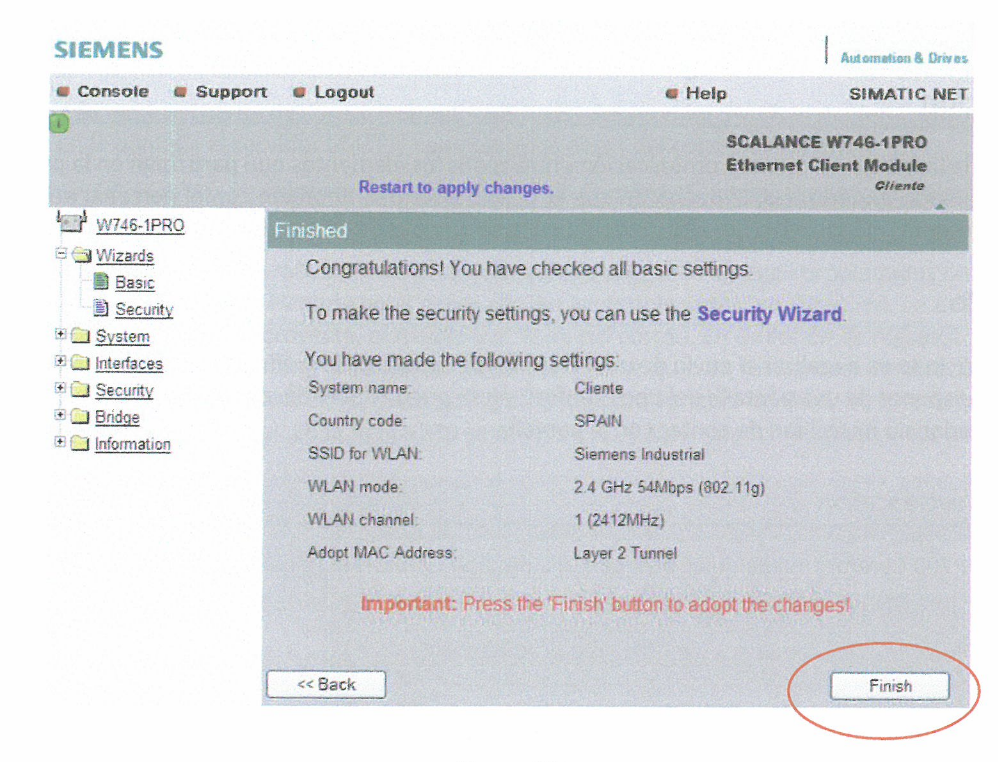


Figura 337

Si se realiza el WIZARD de seguridad (algo necesario y obligatorio en la vida real), resulta adecuado utilizar autenticación Open System y encriptado con clave de cifrado AES.

Se pueden ver las MAC de las ET a las que se conecta el Cliente desde el Punto de acceso en el menú LLDP.

Nota de Siemens

Busque en los menús de los equipos el protocolo LLDP. Se ha de comprobar que se encuentra habilitado para RX&TX para sus puertos Ethernet, aunque esto debería venir por defecto. Permitirá llevar a cabo un descubrimiento de topología; por ejemplo, a la hora de editar estaciones Ethernet y protocolo ISO seleccionado.

De todos modos, no recomendamos equipos sin Rapid roaming o IPCF para hacer Profinet IO. No nos hacemos responsables del mal rendimiento de la aplicación.

24. Envío de *e-mails* desde HMI (WinCC)

Introducción

En la era de la industria 4.0 la intercomunicación entre todos los elementos que participan en la producción adquiere una gran importancia. El hecho de que el propio sistema interactúe con el personal de mantenimiento representa un indicativo del alto grado de eficacia que puede alcanzarse con estos medios técnicos.

Enunciado

En este ejercicio se va a realizar el envío de un *e-mail* desde una pantalla gráfica, de una forma muy sencilla. Se necesita disponer de una pantalla del tipo Comfort; no se puede hacer desde una Basic. Se puede realizar con el simulador sin necesidad de contar con la pantalla.

Material necesario

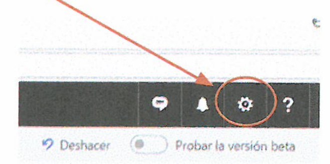
1. Pantalla TP700 Comfort (o cualquier otra pantalla del tipo Comfort si se hace a nivel de simulación).
2. Conexión a Internet.

Realización

En primer lugar, se debe disponer de un servidor SMTP (envío de correos). En este ejercicio se va a realizar desde el servidor de Hotmail, aunque se darán también los datos para GMX.

Los servidores SMTP convencionales solo permiten enviar *e-mails* si proceden de fuentes seguras. Las pantallas de tipo Comfort y las CP pueden utilizar estos servidores (Hotmail, Gmail o GMX). Las CPU sin CP no pueden y tienen que utilizar servidores locales propios de la empresa.

Se ha de disponer de una cuenta de correo de Hotmail. Una vez creada, se abre la cuenta y se va a las opciones de configuración (**desde la rueda dentada**).



Una vez allí, se debe ir a *Opciones* y, en la opción de *POP/IMAP*, se observa el nombre del servidor SMTP y el puerto. Estos datos se necesitarán posteriormente en la pantalla (figura 338).

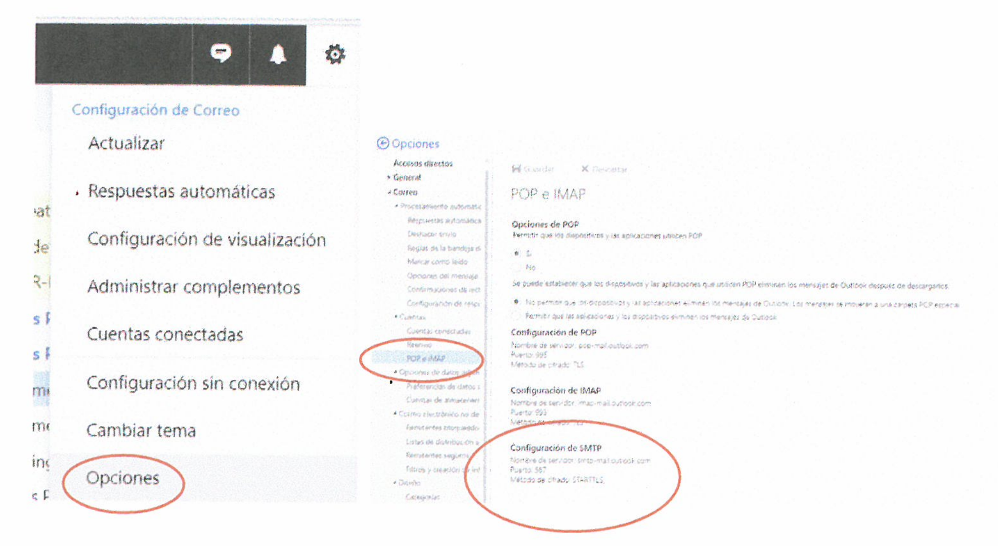


Figura 338

Se observará que el nombre del servidor SMTP es «smtp-mail.outlook.com» y el puerto de salida, el 587. A esos datos habrá que añadir la opción de necesidad de conexión segura, como se verá posteriormente.

Una vez que se tiene la cuenta de correo Hotmail y los datos del servidor SMTP, el siguiente paso consiste en configurar en la pantalla el servidor SMTP y completar los datos del correo de destino.

Ahora, desde TIA PORTAL, se selecciona la pantalla TP700 Comfort. Si no se dispone de dicha pantalla, se puede simular cualquiera de las Comfort.

Se añade un *Botón* a la imagen raíz, se pone el texto *Enviar email* y se le agregan propiedades de *Eventos*. Se añade la función *EnviarCorreoElectrónico* al hacer clic. En las propiedades, en *Dirección,* se debe indicar la dirección donde va dirigido el proyecto, el *Asunto* y el *Texto* del correo. En *Dirección de respuesta,* se escribe la dirección a la que se contestará, si se desea. Debe observarse que, en los campos de dirección, texto y asunto, se puede colocar el texto en formato String o en formato variable de HMI. En este caso se va a introducir el texto directo (String). En la figura 339 se muestra la imagen raíz con el botón y sus propiedades.

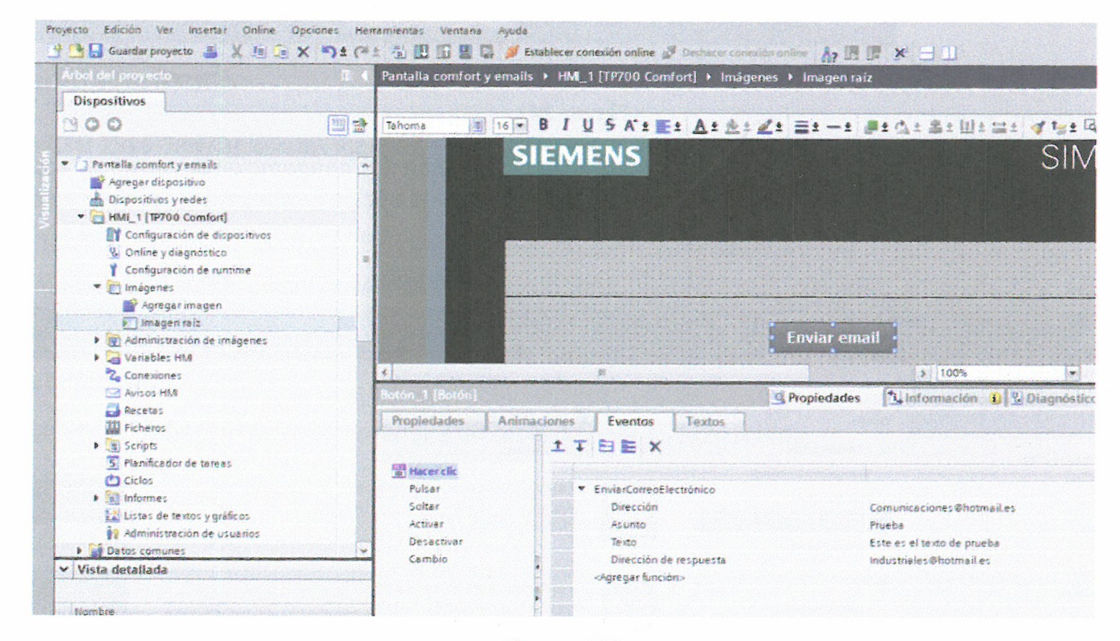


Figura 339

Ahora se va a configurar el servido SMTP (figura 340). Para ello, desde el árbol del proyecto, se accede a *Configuración de runtime*.

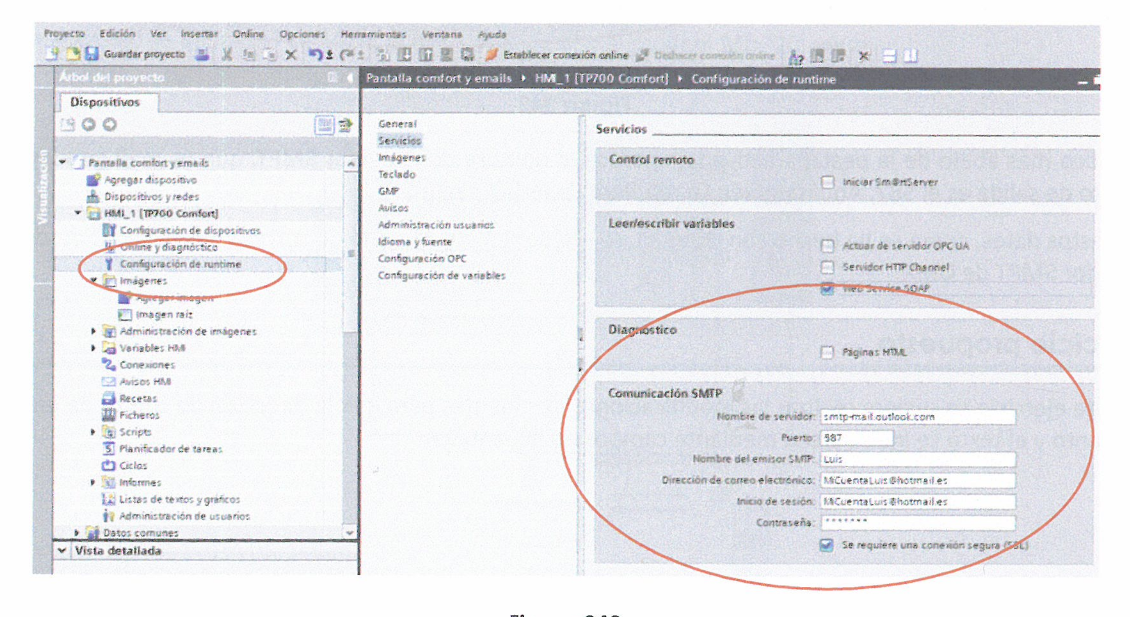


Figura 340

Aquí es donde se debe colocar el nombre del servidor y el número de puerto visto antes. El *Nombre del emisor SMTP* es el que aparecerá en el correo enviado. En la *Dirección de correo electrónico*, se pondrá la dirección de la cuenta Hotmail creada; en *Inicio de sesión*, el nombre de usuario (normalmente la misma dirección de correo) y, en la contraseña, se insertará el *password* de la cuenta de Hotmail. No hay que olvidarse de marcar tampoco la casilla de *Se requiere una conexión segura (SSL).*

Con estos datos, ya se puede arrancar el simulador y activar el botón para enviar el *e-mail*. Si no aparece en la bandeja de entrada, se buscará dentro del correo no deseado.

De esta forma tan sencilla, se pueden enviar correos electrónicos desde una pantalla de tipo Comfort.

Si se desea configurar GMX, habrá que crearse una cuenta en **gmx.es** y rellenar los campos solicitados (figura 341).

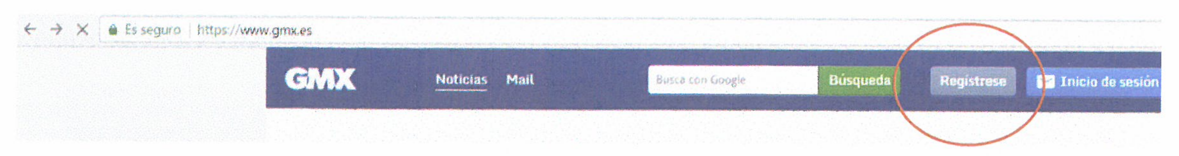


Figura 341

Una vez creada, se entra en el correo y, desde la pestaña *Inicio*, opción de *Configuración de correo*, se accede a la opción POP3/IMAP. En esta ventana resulta importante marcar la opción *Enviar y recibir correos electrónicos a través de programas externos* (figura 342).

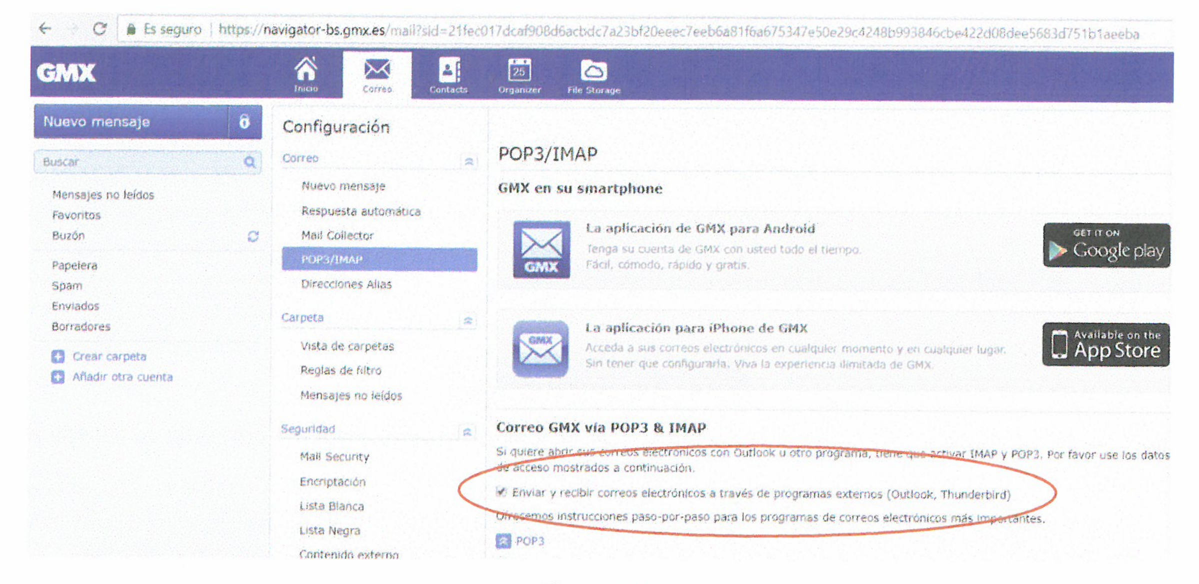


Figura 342

Un poco más abajo de la pestaña activada, aparece el nombre del servidor SMPT, que es **mail.gmx.es.** El puerto de salida es el 587. Aquí también se requiere conexión segura.

Con estos datos, como se ha hecho con Hotmail, se puede realizar el envío de correos electrónicos desde el servidor SMPT de GMX.

Ejercicio propuesto

En este ejercicio se sugiere realizar las modificaciones pertinentes para que la dirección de correo de envío, el asunto y el texto se introduzcan mediante campos en la pantalla.

25. Envío de SMS (SINAUT)

Introducción

Siguiendo con aplicaciones que se encuadran dentro de la industria 4.0, ahora se va a realizar un ejercicio en el que la interconectividad entre la máquina y el personal de mantenimiento resulta inmediata. Constituye un claro ejemplo de efectividad y eficacia. Nadie tendrá que avisar que la producción se ha detenido o que se ha pulsado una emergencia porque el propio sistema pondrá en conocimiento del operario, y de forma instantánea, el problema acaecido.

Enunciado

Se desea realizar un ejercicio con el fin de que se mantenga informado a un operario a través del teléfono móvil mediante un mensaje de texto: SMS.

Material necesario

1. Una CPU 1516-3PN/DP con los módulos digitales y analógicos dispuestos en el resto de ejercicios.
2. Un PC con conexión serie RS232.
3. Un módulo SIMATIC S7 1500, CM PTP RS232 BA (6ESS7 540-1AD00-0AA0).
4. Un módem SINAUT ST7, MD720 2G GSM/GPRS.
5. Una antena, ANT 794-4MR GSM.
6. Una tarjeta SIM.
7. Un cable de Profinet para las conexiones y un cable de serie RS232 estándar.
8. Programa PuTTY.

Realización

Para realizar este ejercicio, se necesitará un módulo adaptador vía serie RS232 entre el PLC y el módem. Este será el encargado de pasar los parámetros y de enviar y recibir los mensajes de texto (SMS) al módem MD720. Este módulo es el CM PtP RS232 BA (6ESS7 540-1AD00-0AA0), de Siemens. También podría utilizarse el CM PtP RS232 HF. Este módulo establece una comunicación punto a punto con el PLC.

En la figura 343 se muestran las diferentes configuraciones que se dan para el envío y la recepción de SMS. En este ejemplo se utiliza el PLC 1516, el módulo CM PtP y el módem MD720.

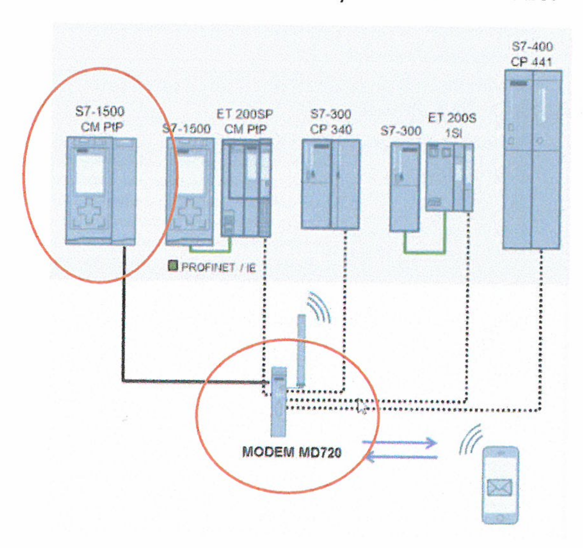


Figura 343

Configuración de PLC y CM PtP

El proceso comienza con la creación del proyecto y la inclusión del módulo CM PtP. Si no estuviera en el catálogo, se deberá actualizar por los procedimientos conocidos. En la figura 344 se observa cómo aparece incluido como un módulo más del PLC. Pulsando sobre el dispositivo CM PtP, se configurará desde sus propiedades según los siguientes parámetros (figura 345):

- Protocolo: Freeport.

Parámetros del puerto de comunicaciones:

- Velocidad: 9.600 bits/s.
- Paridad: no.
- Bits de datos: 8.
- Bits de parada: 1.
- Control de flujo: no.
- Tiempo de retardo de caracteres: 1.000 *bit times* (figura 346).

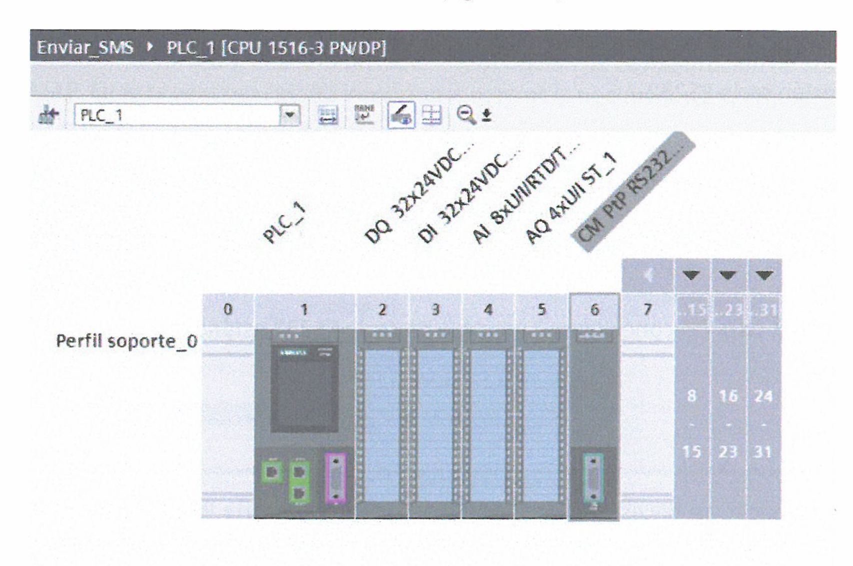


Figura 344

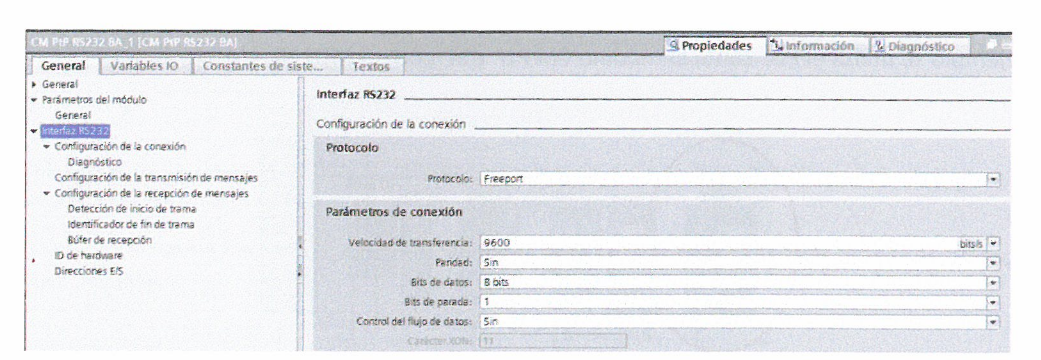


Figura 345

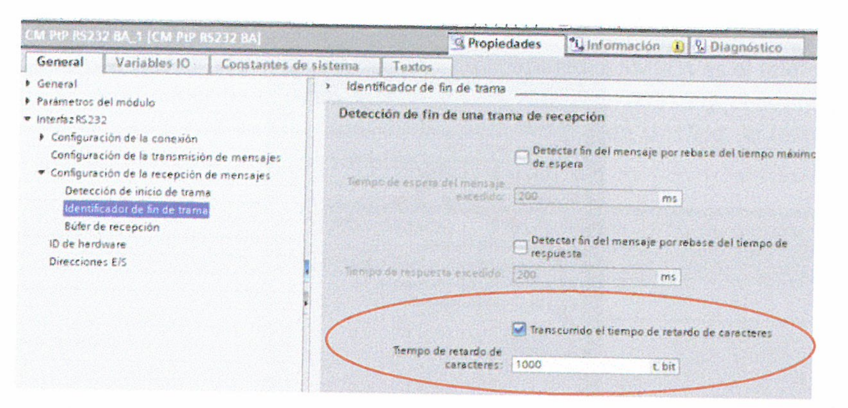
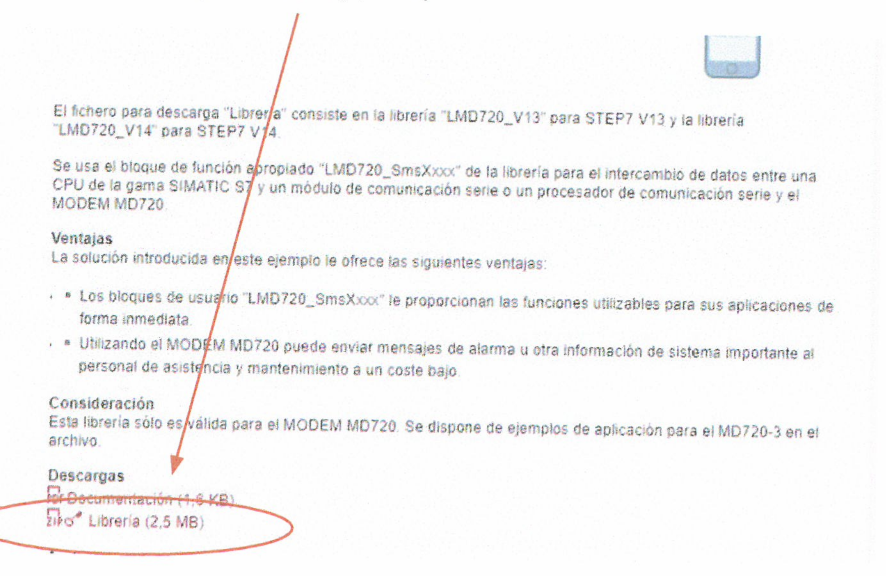


Figura 346

Una vez completada la configuración *hardware,* se debe preparar la parte de *software*. Lo primero debe ser bajarse la librería del módem, MD720, que se introducirá en la programación. Dicha librería se descarga de la web de Siemens. Con la siguiente dirección se accede directamente (se debe recordar que se ha de estar registrado) a la librería e incluye la versión V13 y V14 de TIA PORTAL:

https://support.industry.siemens.com/cs/document/25545680/
envío-y-recepción-de-mensajes-sms-con-cp-o-cm-serie-y-el-módem-md720?dti=0&lc=es-WW

Una vez en la página, se descarga la **librería** (2,5 MB).



Tras descargarse por completo, se incluye en el proyecto. Para ello, se descomprime la versión deseada en una carpeta. Desde TIA PORTAL, se accede a *Librerías globales*, desde el área derecha del portal y en la pestaña de *Librerías*. Allí se activa *Abrir librería global* y se busca la librería descargada. A continuación, aparecerá la nueva librería MD720. Esto puede verse en la figura 347.

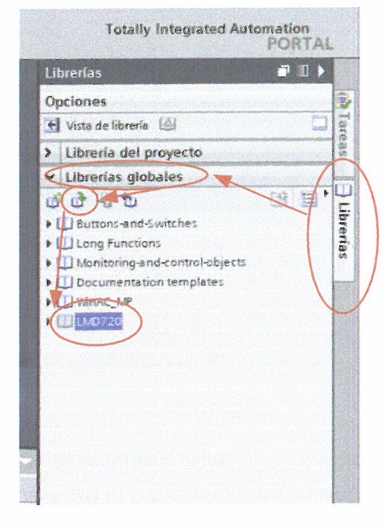


Figura 347

Ahora se abre el OB1 y se arrastra hasta allí la librería *LMD720_SmsCMPtP,* situada dentro de *Tipos* y de *S7-1500* (figura 347).

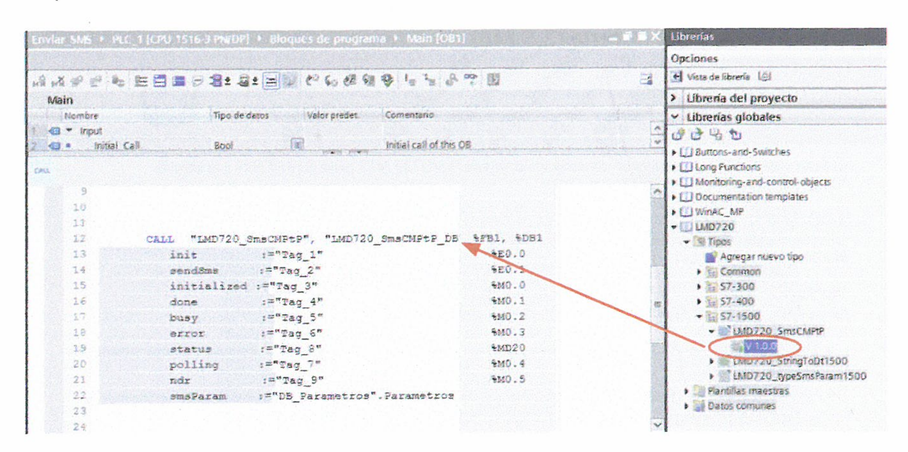
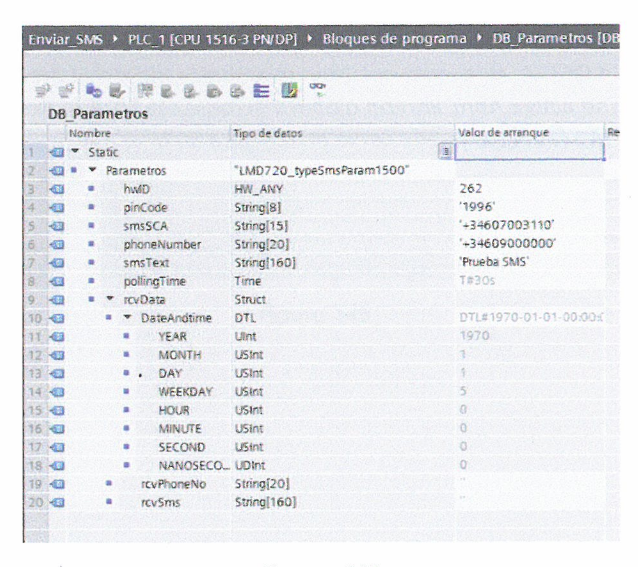


Figura 348

A continuación, se debe crear un *DB global,* así como un parámetro estático del tipo *LMD720_typeSmsParam1500* (figura 349). Si en algún momento sale que *Ha cambiado la interfaz*, se actualizará con el icono (actualizar interfaz). Además, se completan los parámetros que aparecen en el *DB global*.


Enviar_SMS ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Bloques de programa ▸ DB_Parametros [DB

DB_Parametros

	Nombre	Tipo de datos	Valor de arranque
1	Static		
2	Parametros	"LMD720_typeSmsParam1500"	
3	hwID	HW_ANY	262
4	pinCode	String[8]	'1996'
5	smsSCA	String[15]	'+34607003110'
6	phoneNumber	String[20]	'+34609000000'
7	smsText	String[160]	'Prueba SMS'
8	pollingTime	Time	T#30s
9	rcvData	Struct	
10	DateAndtime	DTL	DTL#1970-01-01-00:00:0
11	YEAR	UInt	1970
12	MONTH	USInt	1
13	DAY	USInt	1
14	WEEKDAY	USInt	5
15	HOUR	USInt	0
16	MINUTE	USInt	0
17	SECOND	USInt	0
18	NANOSECO...	UDInt	0
19	rcvPhoneNo	String[20]	
20	rcvSms	String[160]	

Figura 349

El parámetro HWID se encuentra en *ID de hardware* dentro de las propiedades del módulo CM PtP (figura 350). En *pinCode,* se debe colocar el pin de la tarjeta SIM. El parámetro SMSCA es el teléfono del centro de mensajes de texto del distribuidor de telefonía móvil (Vodafone +34607003110). El número de teléfono al que se dirige el mensaje se inserta en el parámetro *phoneNumber* y, en *smsText*, el texto del SMS; todos estos parámetros siempre van entre comillas simples, excepto HWID.

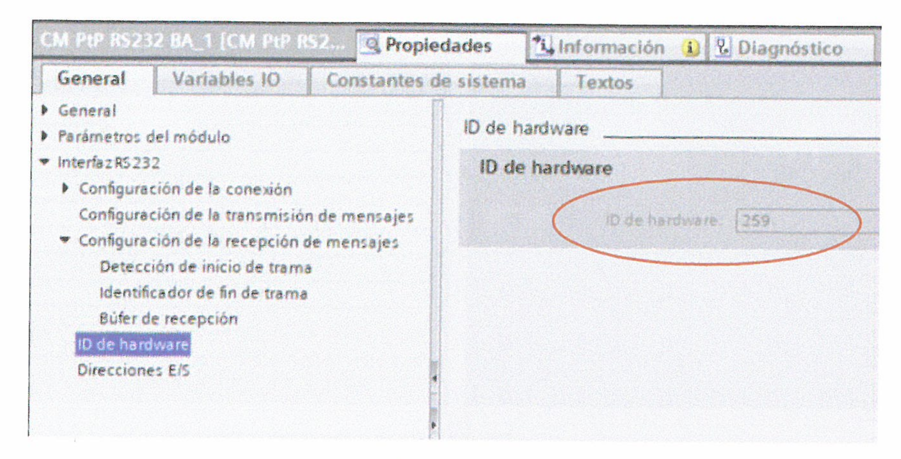


Figura 350

Por último, se deben completar los parámetros de la FB en el OB1, como se puede ver en la figura 351. En el parámetro *init,* se inserta poner la entrada de bit que inicializa el módem. En *sendSms,* se coloca la entrada que activa el envío del SMS. En *smsParam* ha de ponerse el parámetro estático creado en el *DB global*. Se puede escribir o arrastrar desde el DB, como se observa en la figura 351. Para arrastrar, se divide el área del editor en dos ventanas, en **forma vertical.**

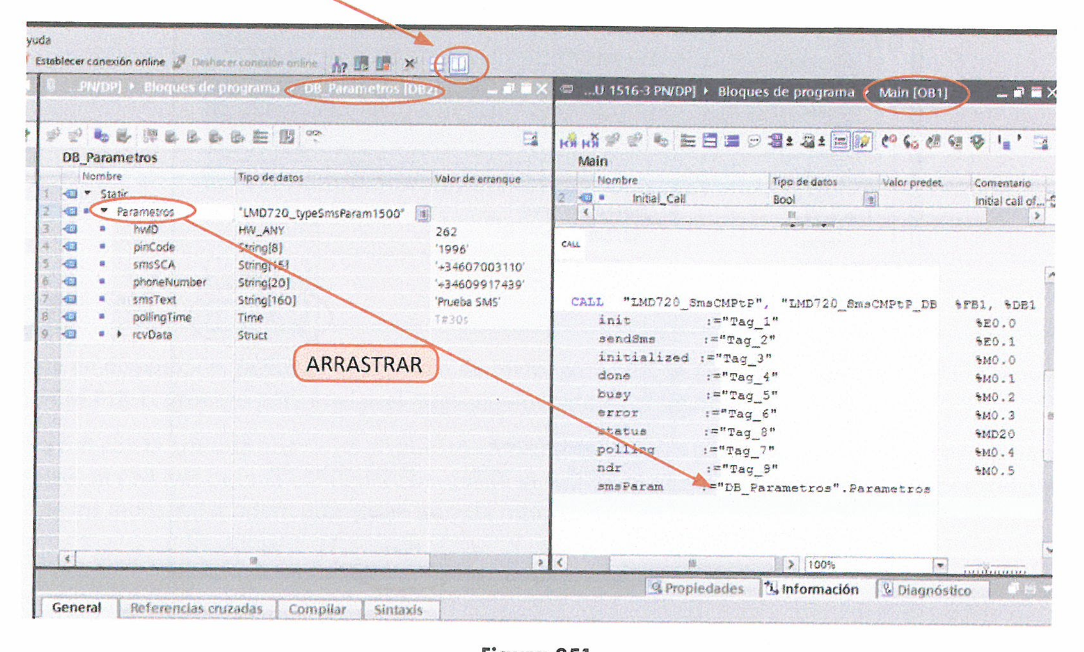


Figura 351

Configuración del módem MD720

Antes de poder enviar mensajes, se ha de preparar el módem MD720. Lo único que habrá que hacer es modificar la velocidad de transferencia del módem de 19.200 bits/s a 9.600 bits/s.

Esto se realiza mediante un comando AT y el PC: se ha de instalar en él un programa de terminal. Este programa puede ser el HyperTerminal u otro. Aquí empleamos en PuTTy, que puede bajarse de Internet, ya que es de uso libre. Un enlace para bajarlo es:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

En primer lugar, se conecta el módem con el PC mediante un cable de serie estándar. Para ello, se necesita que el PC disponga de un conector DB de nueve pines para conexión de serie RS232. No resulta fácil de encontrar hoy día en los PC. Se puede utilizar un cable convertidor de RS232 a USB, pero no se garantiza su funcionamiento para esta aplicación. En la figura 352 se observa un dispositivo de este tipo que se ha comprobado que funciona correctamente. Si no se dispone de dicho conector, habría que recurrir a otro *hardware*, como el SCALANCE M874-3, que puede conectar por Ethernet. Este tema excede el objetivo del ejercicio.





Figura 352

Una vez colocado el cable entre el PC y el módem, se abre el programa PuTTy y se configura la sesión tal como se muestra en la figura 353.

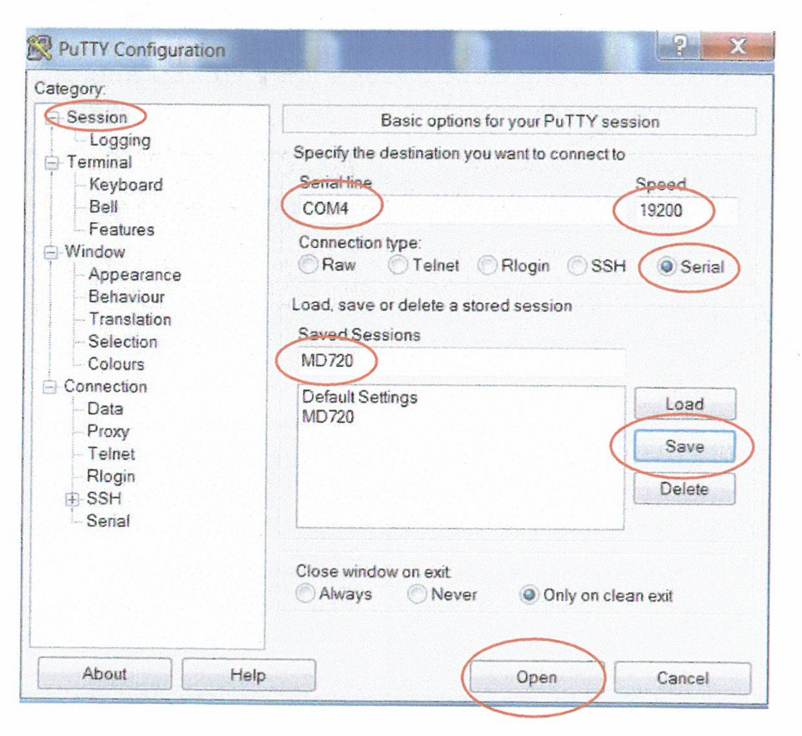


Figura 353

El módem MD720 sale de fábrica con las siguientes características:

Velocidad: 19.200 bits/s.

Bit de datos: 8 bits.

Bit de Stop: 1 bit.

Paridad: sin paridad.

Control de flujo: sin control.

Formado de caracteres: 8N1.

Para volver al estado de fábrica, se acciona durante 4 segundos, o más, el pulsador SET del módem que se encuentra en el frontal.

Se activa la opción *Serie* y se pone el número COM utilizado en el PC, de modo que se coloca la velocidad del módem (19.200 bits/s). Se puede encontrar el COM empleado en el PC en la carpeta *Sistema/Administrador de dispositivos* del panel de control. Se recomienda otorgar un nombre a la sesión con el fin de grabarla y no tener que volver a introducir estos datos cada vez que se utilice.

A continuación, se abre la consola de comandos AT y, en primer lugar, se introduce el pin de la tarjeta SIM. Una vez enviado ese parámetro, se espera a que el módem tenga solo encendido intermitente (1 s de cadencia) el led *Status*. Después, se cambia la velocidad del módem a 9.600 bits/s. En la figura 354 se indica el parámetro utilizado. Cada vez que se introduce un comando AT, el módem responderá con un OK. Si no ocurriese así, es que no habría comunicación.

Tras realizar esto, el estado de los bits debe mostrarse del siguiente modo (se ha de esperar unos segundos):

- Status: fijo encendido.
- Quality: fijo encendido.
- Connect: intermitente.

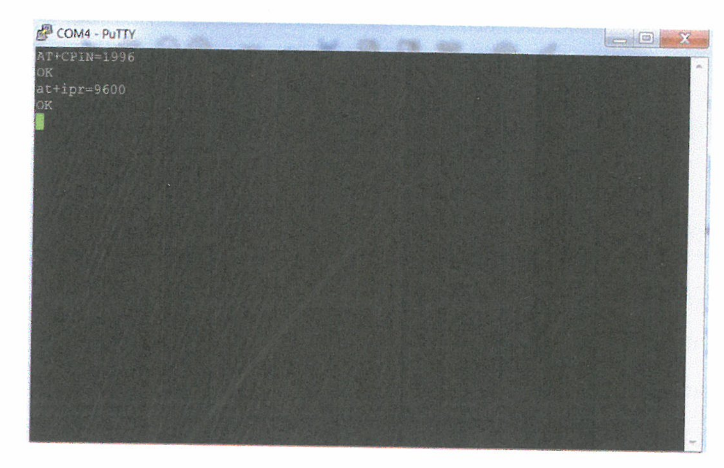


Figura 354

Envío del SMS

Ya se está en disposición de enviar el mensaje de texto; para ello, se debe quitar el cable del PLC y pasarlo al módulo CM PtP. El proceso para enviar el mensaje de texto es el siguiente:

Se activa *Init* con la entrada que se ha dispuesto en dicho parámetro (actúa con el flanco de subida). En estos momentos, el CM PtP pasa la configuración desde el PLC 1516 al módem MD720. Se espera tres minutos hasta que los ledes del módem aparezcan de este modo:

- Status: apagado.
- Quality: encendido fijo.
- Connect: intermitente.

Una vez los diodos led permanezcan así, se podrá confirmar el envío del SMS activando la entrada ubicada en el parámetro *sendSms*. Se podrán enviar más mensajes sin necesidad de volver a inicializar *(Init)*; esto solo se hace una vez.

26. IoT2040 Configuración

Introducción

Si hablamos de la industria 4.0, del Internet de las cosas (IoT), se debe destacar el dispositivo de Siemens IoT2000. Actualmente se comercializan SIMATIC IoT2020 y SIMATIC IoT2040. Constituye la solución de Siemens para conexión a la nube (Cloud). Están equipados con el microprocesador Intel Quark x1020. El módulo se amplía con Arduino Shields y miniPCIe. Resulta asimismo compatible con el sistema operativo Yocto Linux.

Son dos las principales aplicaciones que se pueden realizar con este: la primera, como plataforma de programación de código abierto y de alto nivel. Se debe tener en cuenta que realmente el módulo IoT2040 es un Arduino. Otra importante utilización reside en su uso como pasarela entre el sistema de producción automático y la nube. Siemens ha desarrollado MindSphere como nube para aplicaciones industriales. Se están implementando igualmente aplicaciones específicas que correrán en esa nube. Hoy día, este asunto está todavía por descubrir y desarrollar. El uso de la nube en sistemas industriales resulta de una gran importancia, ya que permitirá evaluar gran cantidad de datos, accesibles así desde cualquier lugar; por ejemplo, el mantenimiento preventivo se beneficiará de este tipo de aplicaciones basadas en la nube.

Enunciado

En este ejercicio se va a utilizar el IoT2040 como plataforma de programación de código abierto. Concretamente, se empleará el lenguaje C++ en la plataforma Eclipse. Se realizará el ejercicio de activar una salida mediante un pulsador.

Material necesario

1. Una pasarela SIMATIC IoT2040.
2. Una fuente de alimentación entre 9 y 32 V DC.
3. Un PC con entrada para tarjetas MicroSD (o adaptador para tarjetas).
4. Una tarjeta Micro SD, de capacidad entre 8 y 32 GB.
5. Un cable de Ethernet.
6. *Software* Eclipse.
7. Imagen de ejemplo formateada al sistema operativo Yocto Linux cargada en la tarjeta MicroSD (proporcionada por Siemens).
8. *Software* PuTTY.
9. *Software* Win32 Disk Imager.

Realización

Antes de describir el proceso del ejercicio, se van a describir las características de la pasarela. En la figura 355 puede verse el IoT2040.



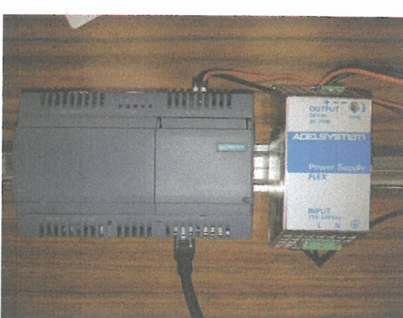


Figura 355

En la figura 356 se contempla el interior del IoT2040. La parte izquierda es igual que un Arduino y, en la derecha se coloca la MicroSD.

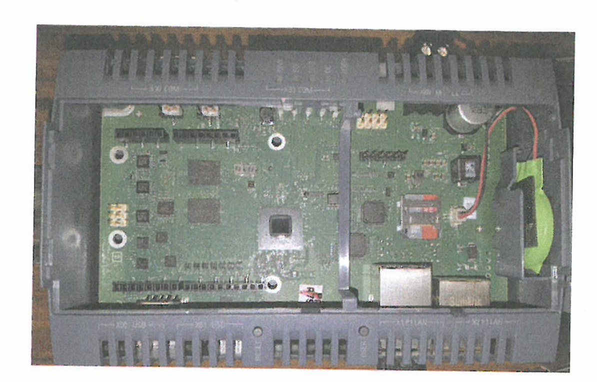


Figura 356

Las características de la pasarela IoT2040 son:

- Procesador de ahorro energético Intel Quark e interfaces integradas: Intel Quark x1020 (+Secure Boot), 1 GB RAM, 2 x Ethernet, 2xRS232/485 y *battery-backed* RTC.
- Soporte de lenguajes Yocto Linux.
- Ampliable con módulos Arduino y tarjetas miniPCIe.
- Programable desde varios lenguajes de alto nivel.
- Diseño compacto y montaje sobre carril DIN.
- Calidad SIMATIC: robusto, fiable y duradero.

Siemens también ha sacado la pasarela IoT2020 con alguna diferencia con IoT2040. Entre otras cosas, la memoria alcanza los 512 MB, el microprocesador es el Intel Quark x1000 y dispone de un solo puerto COM y otro para conexión Ethernet. Todo lo demás aparece igual que el IoT2040.

Para poder realizar este ejercicio, se deben seguir los siguientes pasos:

1. Instalación de la imagen en la tarjeta MicroSD, la cual se descargará de la web de Siemens. Para cargar dicha imagen en la tarjeta, se necesita el *software* Win32 Disk Imager.
2. Inserción de la tarjeta en el dispositivo IoT.
3. Conexión del cable de Ethernet entre el PC (tarjeta de red) y el IoT2040.
4. Establecimiento de conexión entre el PC y el IoT2040 con el programa PuTTY.
5. Cambio de la dirección IP del IoT2040.
6. Descarga del IoT2000 SDK y descompresión con 7-Zip.
7. Bajada del *plug-in* IoT2000 Eclipse.
8. Inserción de la plataforma Java.
9. Conexión de Eclipse e IoT2040 y carga del programa.

Instalación de la imagen en la tarjeta MicroSD

Lo primero que debe hacerse es dar formato a la tarjeta MicroSD para que pueda ser utilizada en el IoT2040. El formato ha de ser para el sistema operativo Yocto Linux. Con el fin de facilitar las cosas, Siemens ha creado una imagen para ser bajada de su web y cargada en la tarjeta, desde la siguiente dirección:

https://support.industry.siemens.com/cs/document/109741799/ simatic-iot2000-sd-card-example-image?dti=0&lc=en-WW

En la figura 357 se puede observar el único archivo que hay que bajarse. Siemens modifica las versiones por lo que puede ser que, al acceder a dicho enlace, la versión se muestre diferente a la de la figura. Una vez descargado, se deberá descomprimir.

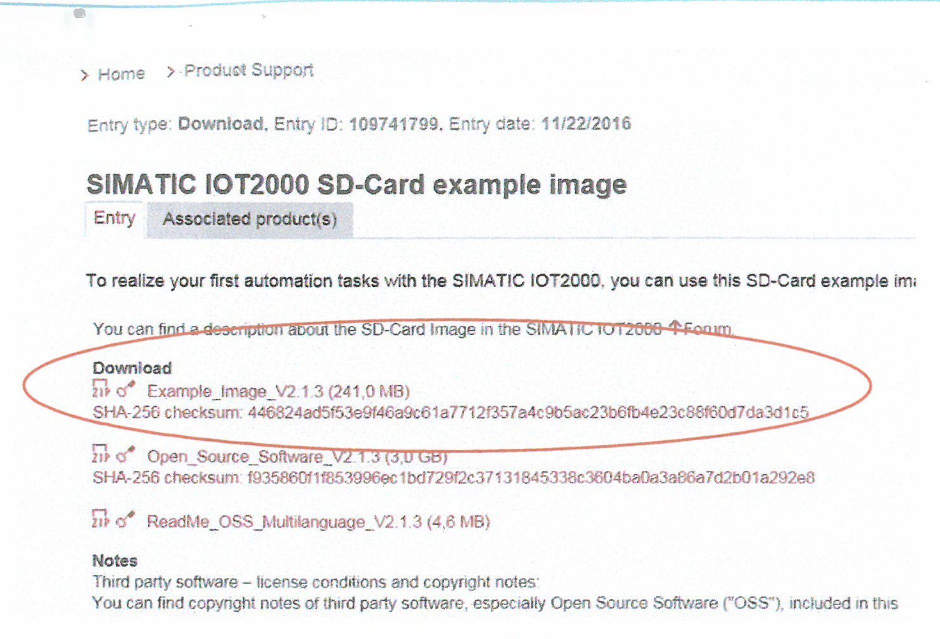


Figura 357

Ahora es necesario bajarse el programa que gestionará la imagen descargada. El programa es Win32 Disk Imager y se baja desde:

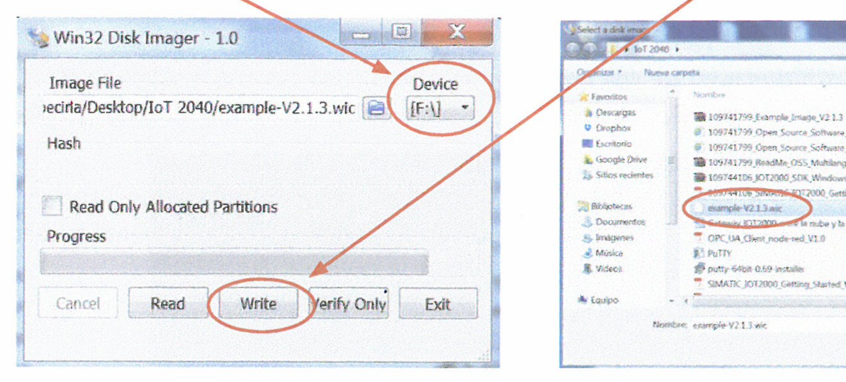
https://sourceforge.net/projects/win32diskimager/

Se coloca la tarjeta en la ranura del PC para tarjetas MicroSD. Si el PC no dispone de esta ranura, habría de conseguir un adaptador de USB para tarjetas SD. En la figura 358 se muestra uno de estos adaptadores.



Figura 358

Se abre el programa Win32 Disk Imager y se busca ( ) el archivo de imagen descargado y ya descomprimido. Para buscar el archivo, se ha de cambiar a buscar todos los archivos, es decir, de **cualquier extensión (*.*).** El archivo debe llevar la extensión **.wic**. Ahora debe indicarse dónde se desea cargar la imagen; esto es, **se selecciona la letra** de la unidad/tarjeta MicroSD. Ahora se activa **escribir**. Se puede apreciar este proceso en las figuras 359 y 360.



Figur 359

Figura 360

Aparecerá un aviso que se debe aceptar y, después se pulsa en OK (figuras 361-362).

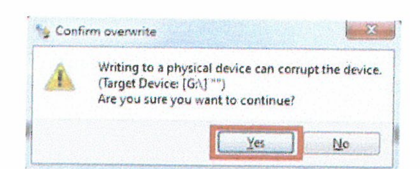


Figura 361

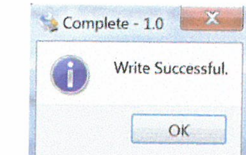


Figura 362

Inserción de la tarjeta en el dispositivo IoT

El proceso de cargar la imagen en la tarjeta ya ha concluido y ahora se debe introducir la tarjeta en el sitio destinado dentro del IoT2040. Se coloca la tarjeta encima del portatarjetas, se cierra y se empuja en el sentido de la flecha. Este paso se aprecia en las figuras 363 y 364.

La primera vez que se inserta la tarjeta con la imagen cargada tarda un tiempo hasta presentarse operativo ya que el IoT2040 realiza una redimensión de la MicroSD. Se debe esperar a que el led SD del IoT2040 se apague de forma permanente.

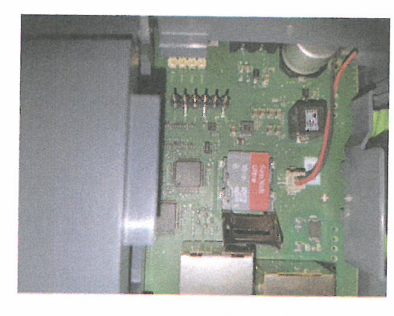


Figura 363

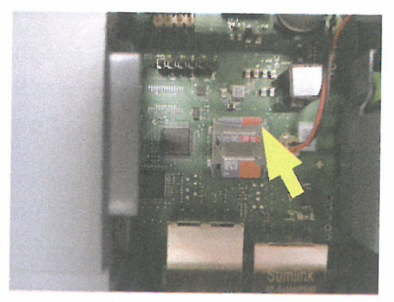


Figura 364

Conexión del cable de Ethernet y establecimieto de conexión con el programa PuTTY

Ahora se debe conectar el cable de Ethernet entre el PC y el IoT2040 para poder conectar ambos. Previamente se ha tenido que enchufar a una fuente de alimentación de entre 9 y 36 V de corriente continua. En el IoT2040, el cable se coloca en *X1 P1 LAN*.

Para realizar dicha conexión, se requiere un programa como el PuTTY, ya utilizado en el ejercicio anterior. Este se puede descargar del siguiente enlace (dependiendo del PC, se bajará el de 32 o 64 bits):

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Una vez abierto el programa PuTTY, se debe poner la dirección de Ethernet del IoT2040 que, por defecto, es 192.168.200.1, el puerto 22 y el tipo de conexión SSH (**S**ecure **SH**ell: intérprete de órdenes seguro). Se le puede dar un nombre a la sesión y salvarla.

Resulta muy importante que la tarjeta de Ethernet del PC cuente con una dirección dentro del mismo grupo de direcciones que el IoT2040. No obstante, como esto no será lo normal, se deberán cambiar las propiedades IPv4 de la tarjeta de red del PC. Se puede dejar como aparece en la figura 365.

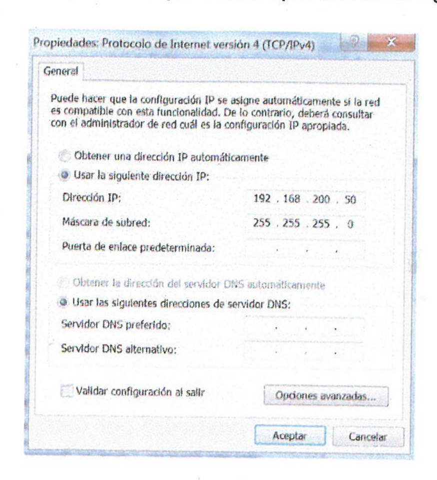


Figura 365

En la figura 366 se aprecia cómo debe quedar el PuTTY para establecer la conexión.

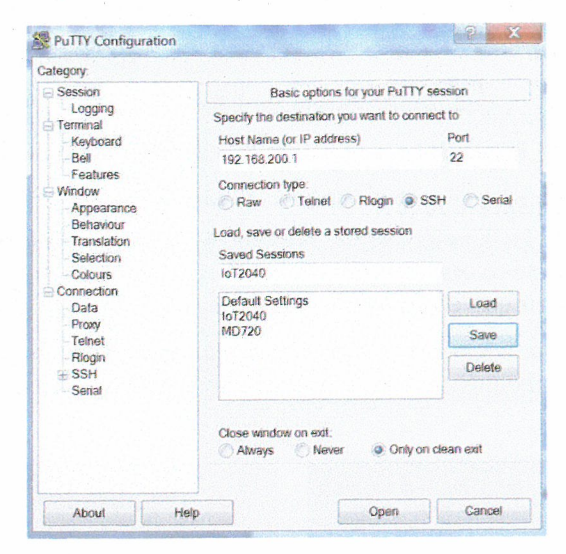


Figura 366

Pulsando en *Open*, se accede al monitor de conexión con el IoT2040. Se puede apreciar en la figura 367 el aspecto inicial. La primera vez que se accede aparece un texto de aviso, donde habrá que pulsar *Yes*, como se observa en la figura 368.

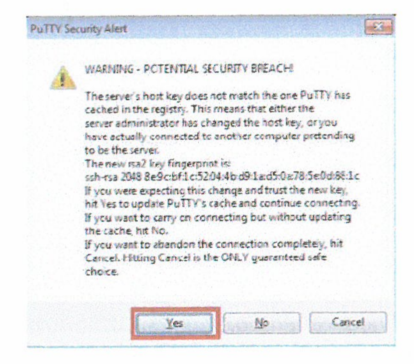


Figura 367

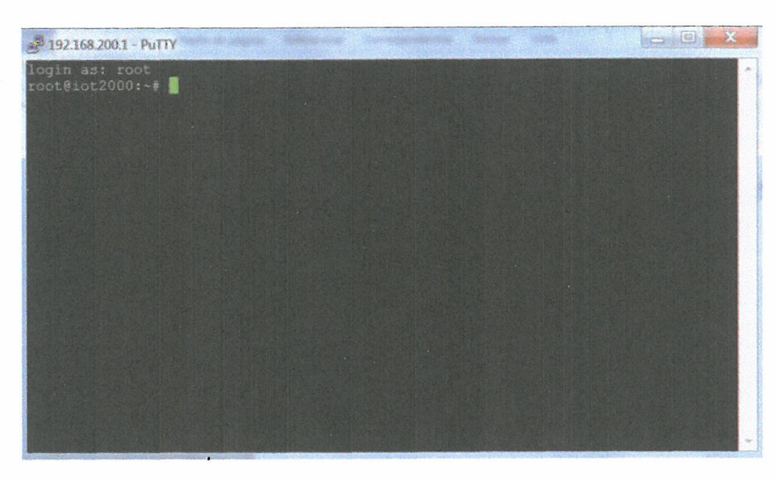


Figura 368

En la pantalla que aparece, se pide el *login*, que es *root* por defecto, y se acepta con *Enter*. En el siguiente paso se debe introducir «*iot2000setup*», que es la *password* por defecto, tal como se aprecia en la figura 369. Si se desea cambiar, se introduce el comando «*passwd*».

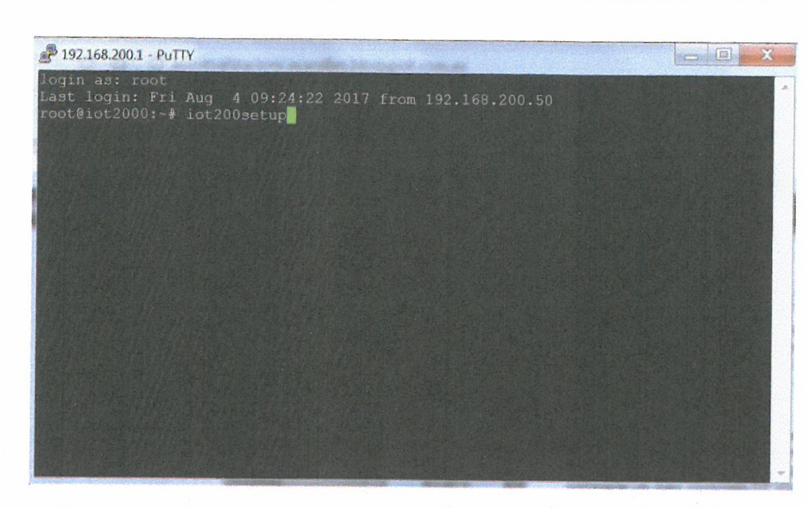


Figura 369

Aparecerá el menú del *setup*, como se refleja en la figura 370.

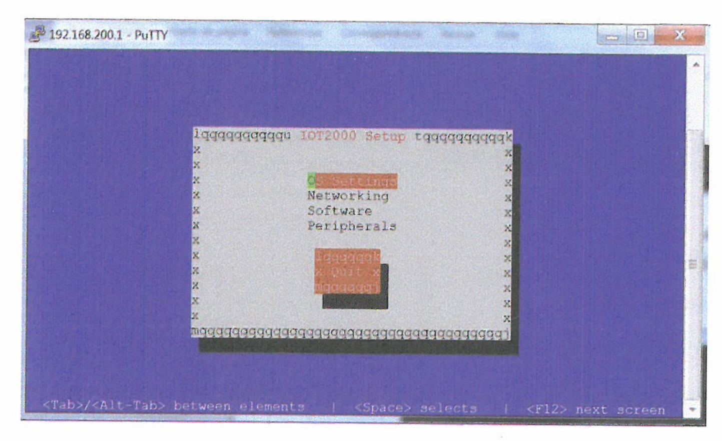


Figura 370

Cambio de la dirección IP del IoT2040

Desde el menú *setup*, se pueden hacer diversos cambios en el IoT2040. Uno de ellos es la modificación de su dirección, con el fin de ponerle una perteneciente al grupo de direcciones de la tarjeta de red del PC. Para ello, se entra en Networking desplazando el cursor con la tecla hacia abajo. Con la tecla de tabulación, se podrá desplazar entre los diversos elementos (*Back*, *Cancel*, *Quit*, etc.). No es obligatorio cambiar la dirección del IoT2040 si no resulta necesario.

Las figuras 371-373 conducen hasta la pantalla de cambio de la dirección.

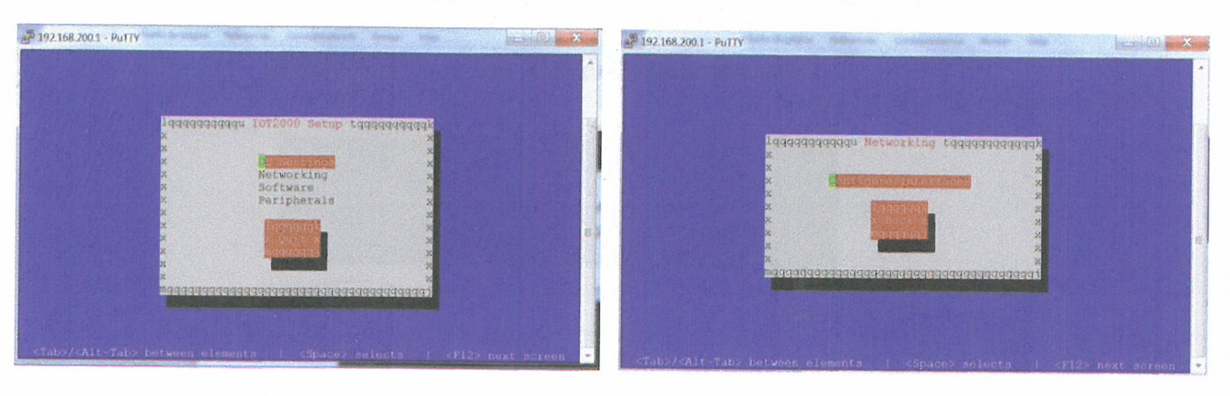


Figura 371

Figura 372

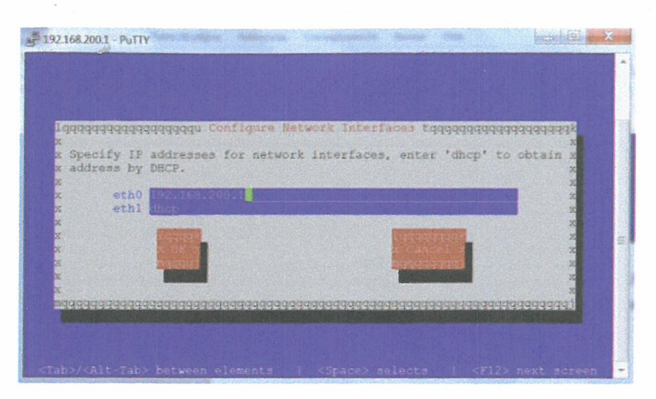


Figura 373

Descarga del IoT2000 SDK y descompresión con 7-Zip, bajada del *plug-in* IoT2000 Eclipse e instalación de la plataforma Java

En este apartado se va a descargar el *software* necesario para que funcione un programa en el IoT2040.

Los programas y complementos necesarios son:

- Complemento Java para el PC, en función del sistema operativo que se dispone.
- Programa Eclipse, donde se realizará el proyecto y se conectará con el IoT2040.
- *Plug-in* del Eclipse.
- IoT2000 SDK.
- Descompresor 7-Zip.

Se bajará Java desde este enlace:

http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

Se deberá seleccionar la opción del sistema operativo que se disponga, por ejemplo, Windows 10 para 64 bits. Se puede tomar el fichero ejecutable para Windows 10, 64 bits, se indica en la figura 374.

Ahora se baja el programa Eclipse, utilizado para programar en C/C++ y realizar una conexión con el IoT2040. Se puede descargar del siguiente enlace:

https://www.eclipse.org/downloads/eclipse-packages/

Se selecciona para Windows de 32 bits o 64 bits.

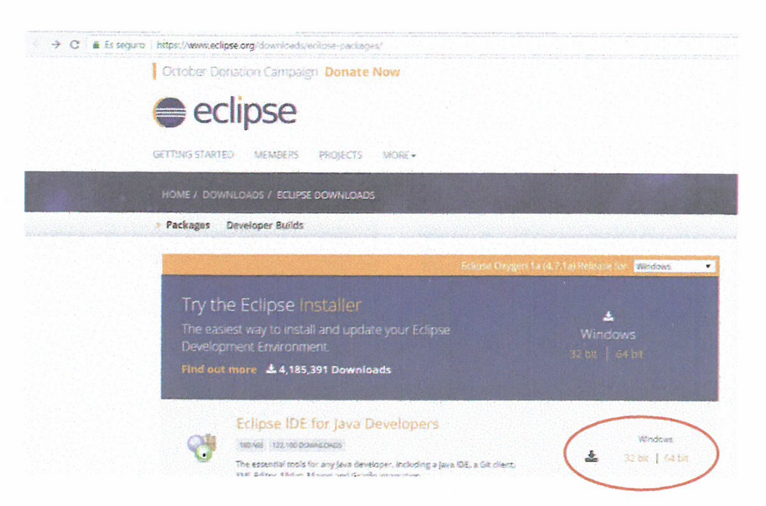


Figura 374

Una vez descargado, se instala y, cuando salga la pantalla de la figura 375, se selecciona la opción *Eclipse IDE for C/C++ Developers*.

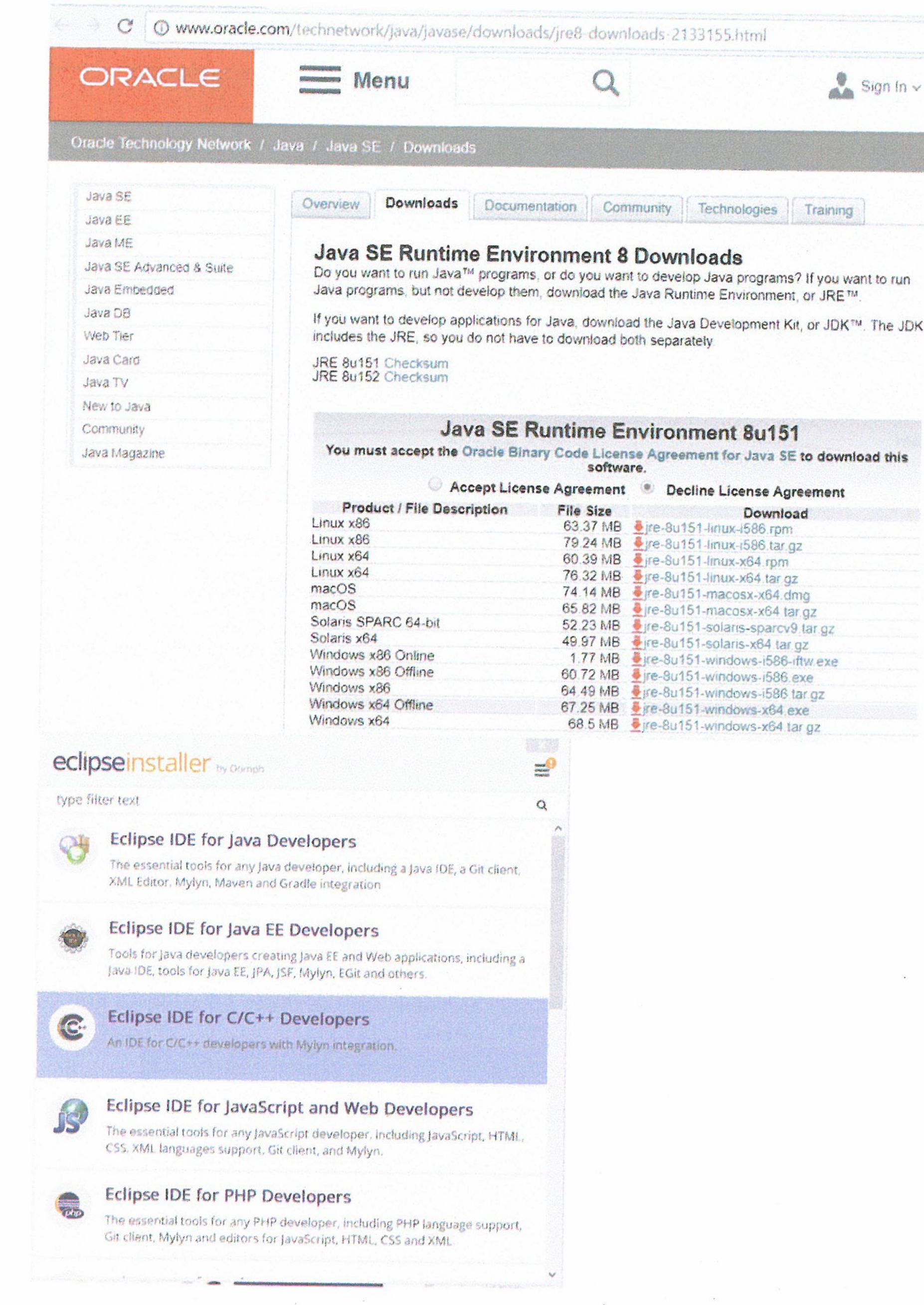


Figura 375

Una vez instalado, se debe bajar el *plug-in* IoT2000 para Eclipse. Con este *plug-in*, se crea la aplicación ejecutable para el IoT2040 y se vuelcan las aplicaciones en el IoT2040. Se puede descargar desde la siguiente página web de Siemens:

https://support.industry.siemens.com/cs/document/109744106/simatic-iot2000-eclipse-plugin?dti=0&lc=de-WW

Desde esa página se accede al **enlace del *plug-in*,** como se aprecia en la figura 376.

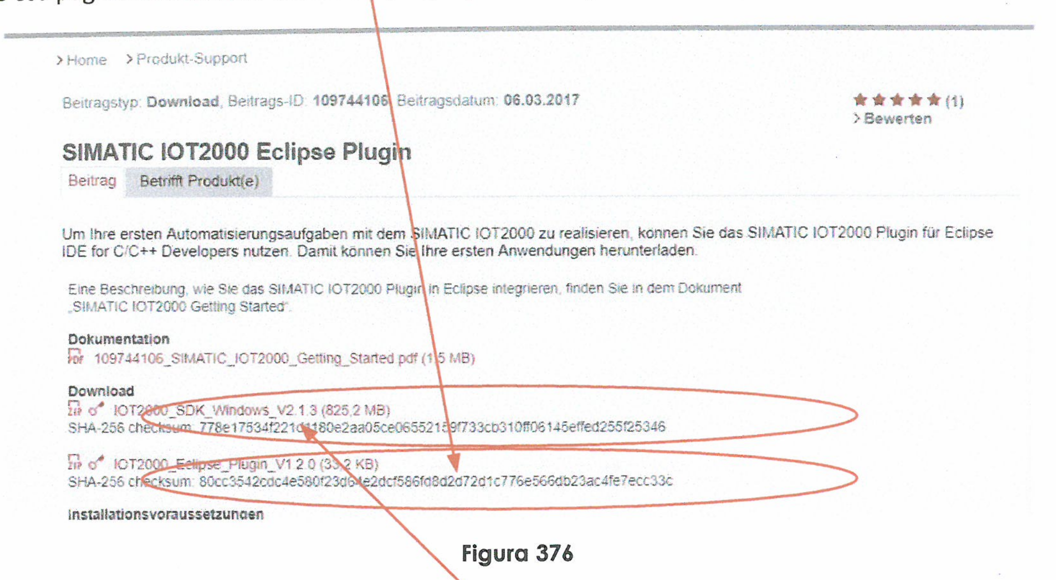


Figura 376

Desde esa misma página se debe descargar el **fichero SDK**, donde se compilan las aplicaciones para el sistema operativo Yocto del SIMATIC IoT2000, necesario para Eclipse y el *plug-in*.

El fichero SDK comprimido lleva una extensión ***.tar***. Para descomprimir este fichero correctamente, se hace con el descompresor 7-Zip (no vale el RAR). Este descompresor es libre y se puede bajar del enlace **http://www.7-zip.org/download.html.** Ahora se debe descomprimir el fichero V2.1.3_SDK_Windows con el 7-Zip y, para que todo salga bien, es necesario hacerlo como administrador. Para ello, se busca el programa instalado y, tocando en el icono del programa con el botón derecho, se selecciona *Ejecutar como administrador*. Se muestra en la figura 377.

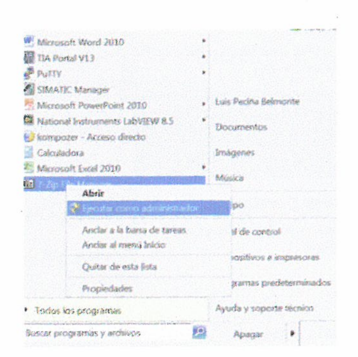


Figura 377

Una vez ejecutado el 7-Zip, como administrador se busca la ruta donde se dejó el archivo SDK descargado, como se aprecia en la figura 378.

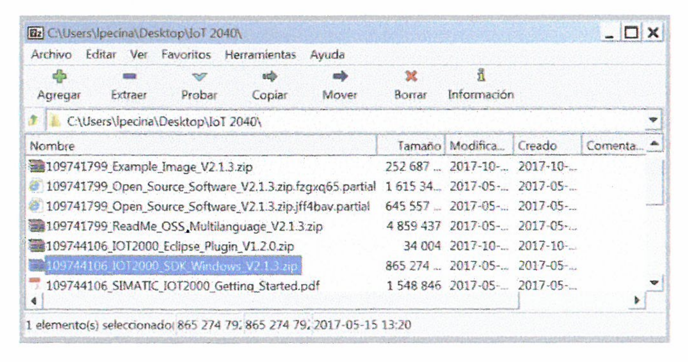


Figura 378

Pulsamos dos veces y aparece un nuevo fichero (figura 379).

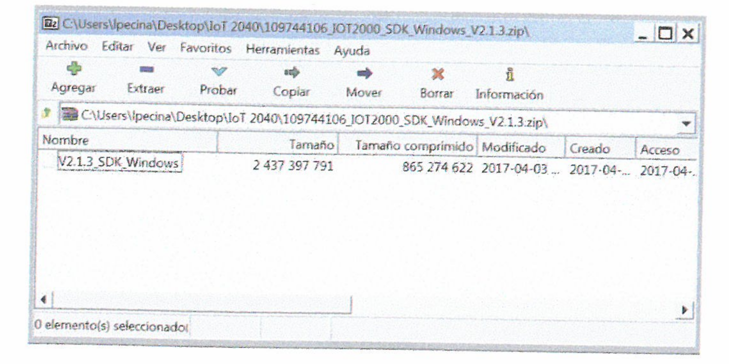


Figura 379

Se vuelve a pulsar dos veces sobre el fichero V2.1.3_SDK_ Windows y se espera hasta que aparezca la figura 380.

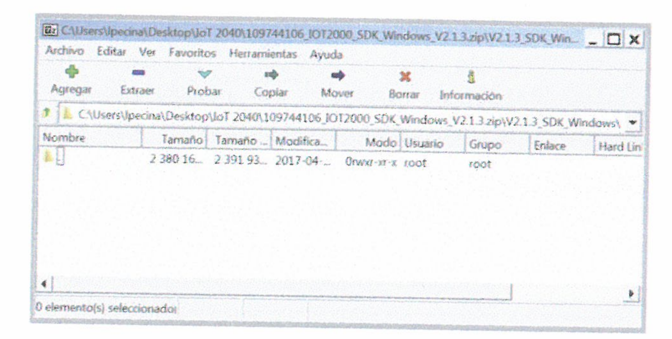


Figura 380

Ahora se vuelve a pulsar dos veces sobre la carpeta que ha salido y se espera a que termine de generar cuatro ficheros y una carpeta, tal como se observa en la figura 381.

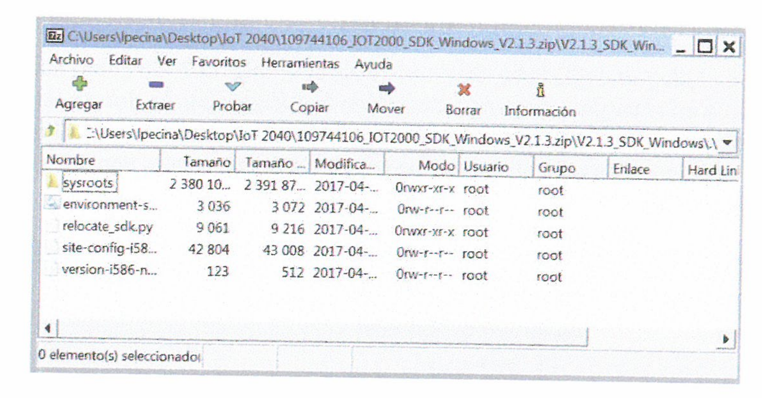


Figura 381

Ahora se seleccionan los cinco elementos y se pulsa sobre *Extraer*. Se dejan en alguna carpeta que ya exista o se haya creado previamente para guardar. Si surge alguna pregunta, se dice *Sí a todo*. Le cuesta bastante tiempo en descomprimir.

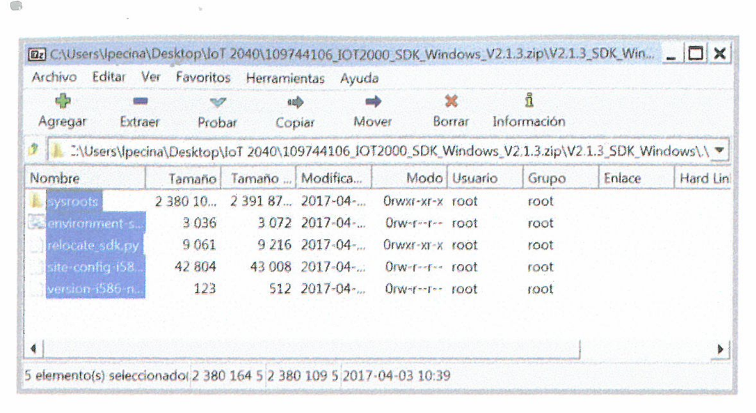


Figura 382

La descompresión será correcta si salen avisos como los de la figura 383; de lo contrario, algo se ha hecho mal. Una vez concluido, se pulsa sobre *Cerrar.*

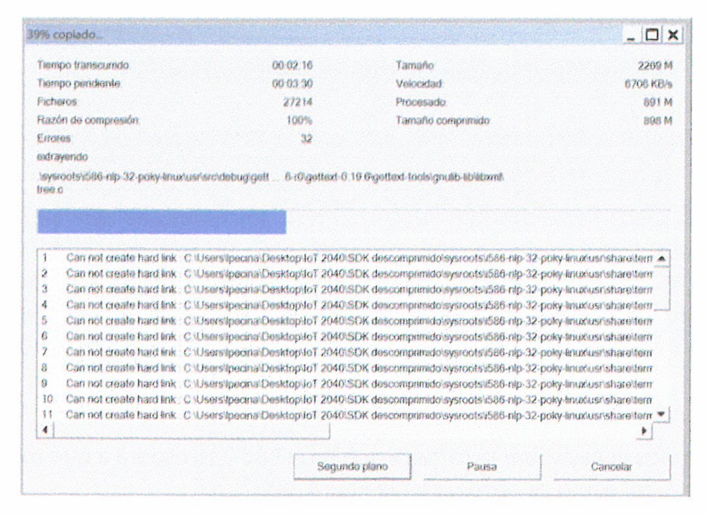


Figura 383

Ahora se descomprime el *plug-in* del Eclipse. Se puede utilizar el mismo 7-Zip como administrador. Una vez descomprimido, se copia en la carpeta *dropins,* ubicada en la misma carpeta donde está cargado el Eclipse. Para acceder a dicha carpeta desde el icono del acceso directo, y con el botón derecho, se selecciona *Abrir ubicación del archivo*. Una vez abierta la ubicación, aparecerá *dropins* y allí se copia el *plug-in* descomprimido. Se muestra en la figura 384.

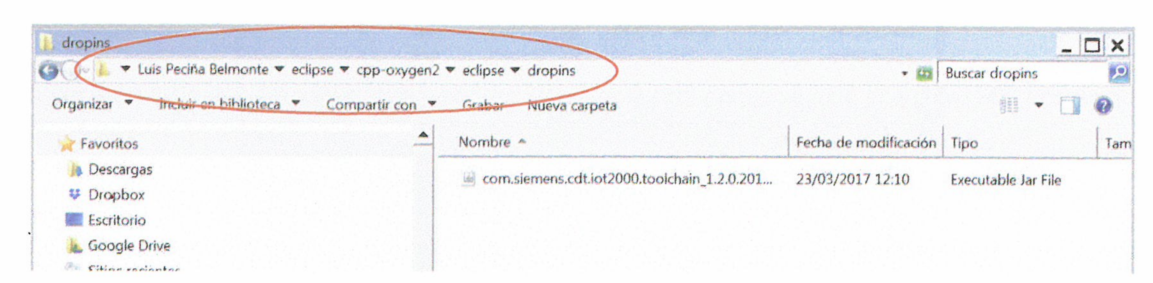


Figura 384

Conexión de Eclipse e IoT2040 y carga del programa

El siguiente paso es realizar la conexión del Eclipse con el IoT2040 y cargar el programa sobre el IoT.

Para ello, se arranca el Eclipse. Se debe modificar el lugar del espacio de trabajo por «*workspace_iot*». Y, a continuación, se pulsa sobre *Launch*. Se aprecia en la figura 385.

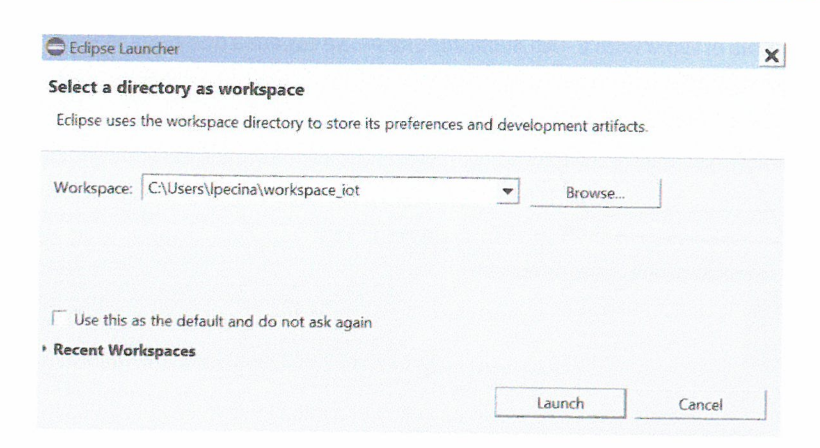


Figura 385

Instalación del TCF

En primer lugar, se instala en el Eclipse el complemento *TCF Target Explorer,* con el que se podrá realizar una conexión con el IoT. Esto se debe hacer solo una vez, no para cada proyecto.

TCF (Target Communication Framework) constituye un protocolo de red extensible, independiente del proveedor, principalmente para comunicarse con sistemas embebidos (integrados). Su principal característica es que TCF ha sido diseñado para conectar de forma transparente servidores entre la herramienta y el objetivo; en este caso, entre el Eclipse y el IoT.

Para instalar el TCF, desde el menú *Help* en la pantalla principal, se va a *Instalar nuevo software*, tal como se observa en la figura 386.

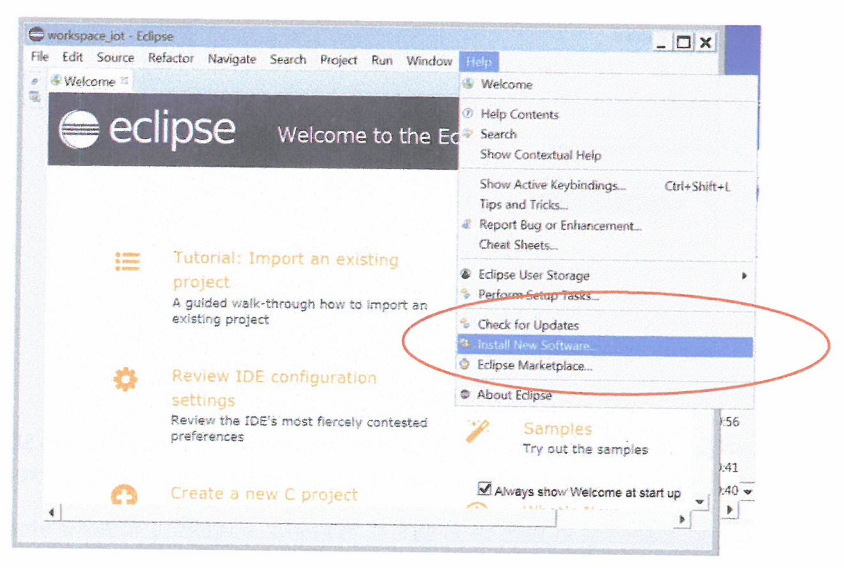


Figura 386

A continuación, se cambia el *Work with* a *--All Available Sites--*. Se espera a que salgan las opciones y se abre **Mobile and Device Development** y, una vez dentro, se selecciona **TCF target Explorer**. Luego, se pulsa sobre **Next** (figura 387).

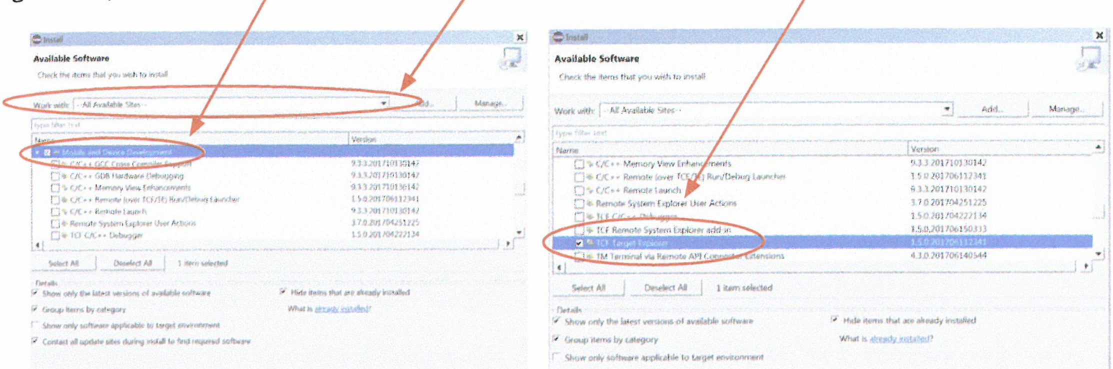


Figura 387

Seguidamente, se marca la opción de aceptar los términos de la licencia y se finaliza. Se cargará el complemento TCF; hay que esperar a que salga una ventana en la que se pida reiniciar el Eclipse y se acepta. El reinicio no se refiere al ordenador, solo al programa Eclipse.

Creación nuevo proyecto en Eclipse

Para crear un nuevo proyecto, se abre el tipo de fichero de SIMATIC IOT2000. Para ello, se va a *File/New/Other*. Se busca en la carpeta *SIMATIC IOT200* y se selecciona *SIMATIC IOT200 Project.* Se observa en la figura 388.

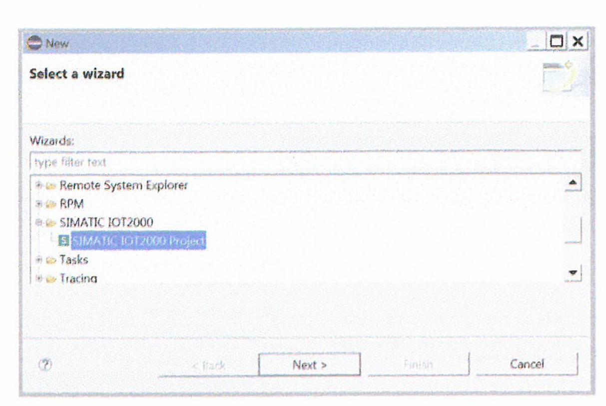


Figura 388

Después de pulsar en *Next,* se debe seleccionar la carpeta donde se descomprimió el fichero SDK. También hay que dotar de un nombre al proyecto que se vaya a crear. Se pulsa en *Finish* para dar por creado el proyecto (figura 389).

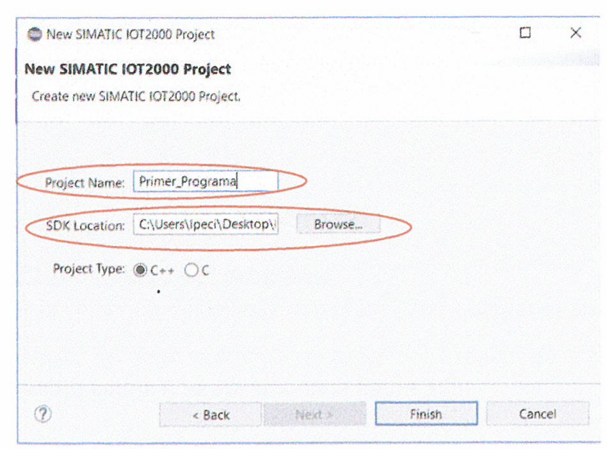


Figura 389

Conexión del IoT con Eclipse

El siguiente paso es hacer efectiva la conexión para cargar el programa en el IoT2040. Para ello, se requiere que exista conexión física, mediante un cable de Ethernet, entre el IoT (conectado a X1) y el ordenador.

Desde el menú *Target,* se crea una nueva conexión, como se aprecia en la figura 390.

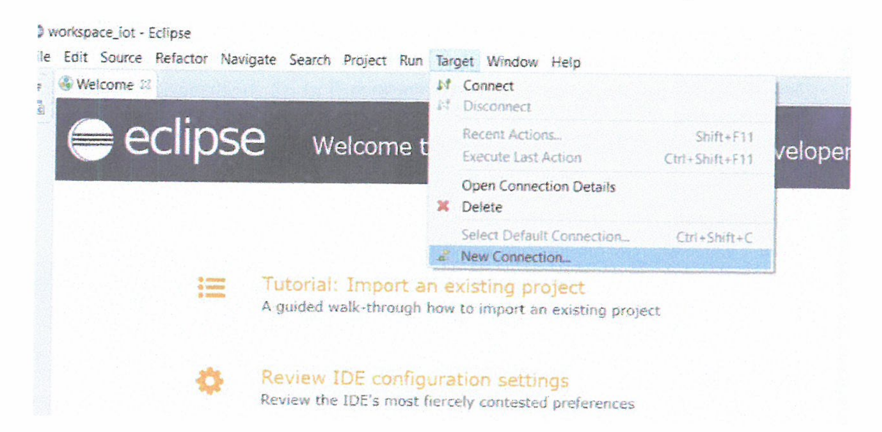


Figura 390

En la pantalla de la figura 391, se selecciona la conexión genérica y se activa *Next*.

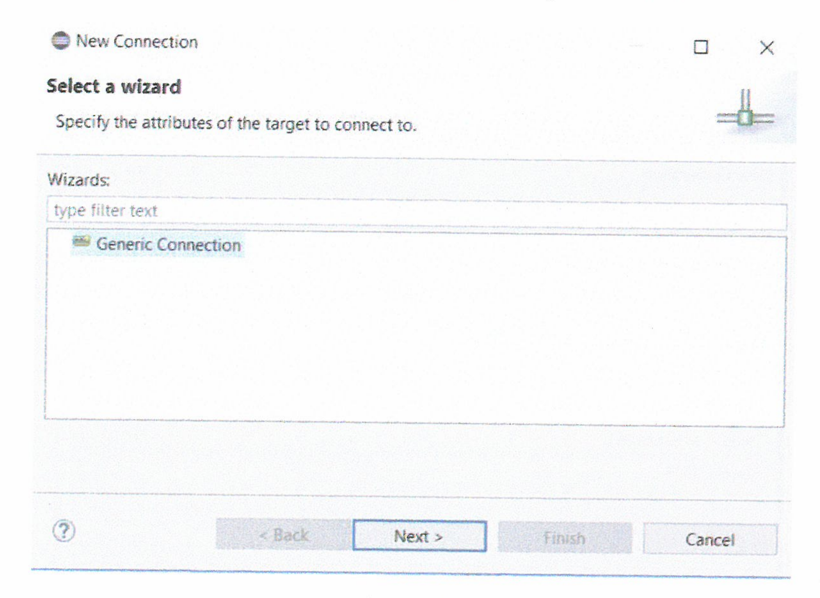
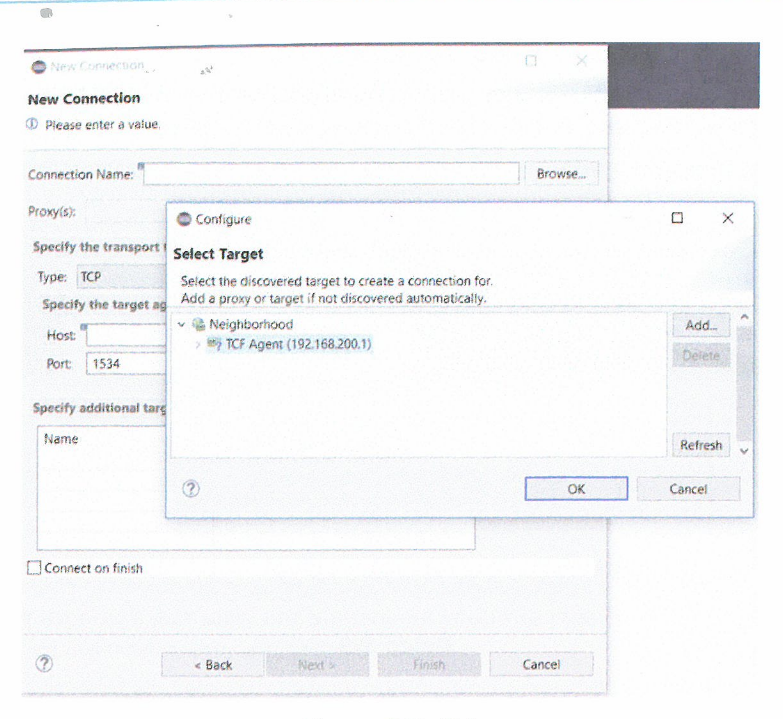


Figura 391

Se busca mediante *Browse* y debería aparecer la conexión con el IoT2040. Si no fuese así, existe algún problema con la dirección del IoT o el Pc. Se debe recordar que deben estar las dos direcciones dentro del mismo grupo (**192.168.200**.X). Si todo va bien, se marca la conexión y se activa *OK*, como se aprecia en las figuras 392-393.



Figuras 392-393

Después de seleccionar, la conexión quedará como se puede ver en la figura 394.

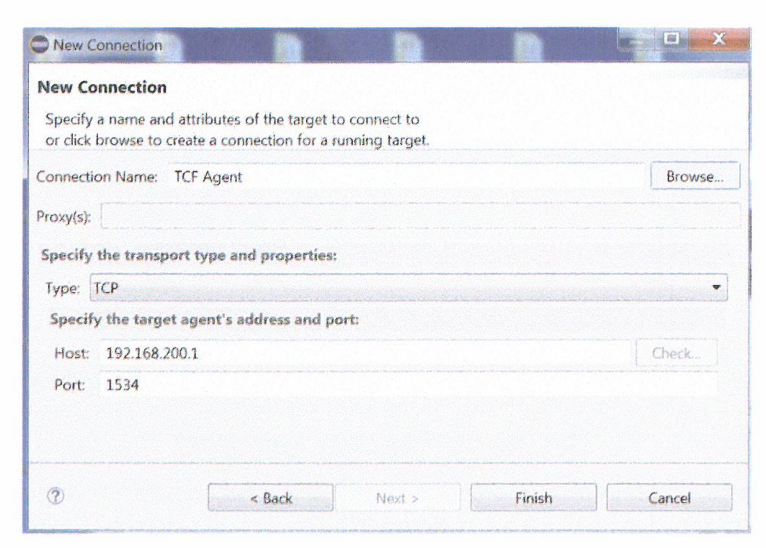


Figura 394

Para cambiar la visión del programa Eclipse, y que salgan todas las herramientas en la barra superior, se activa *Workbench*, como se muestra en la figura 395.

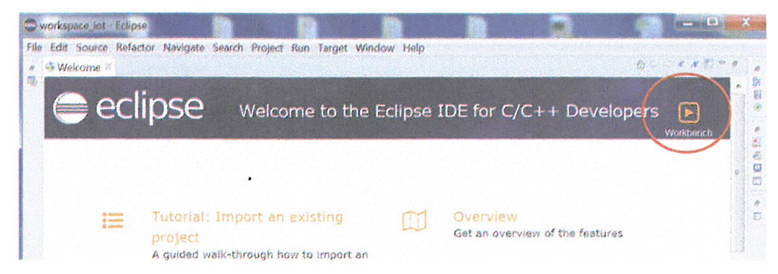


Figura 395

Carga del proyecto en el IoT2040

Seguidamente se va a describir el procedimiento de carga del proyecto sobre el SIMATIC IoT y se podrá visualizar la ejecución del programa sobre el dispositivo.

Desde el nombre del proyecto, y con el botón derecho, se selecciona *Build Project*. Los nuevos archivos aparecerán en el área izquierda de la pantalla, tal como se puede observar en la figura 396.

A continuación, en el mismo menú (botón derecho del ratón sobre el nombre del proyecto), se selecciona *Run As* y, luego, *Remote Application*. En la figura 397 se ilustra lo explicado.

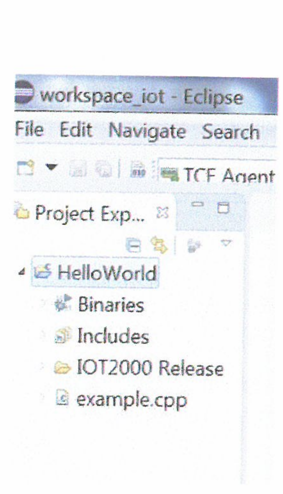


Figura 396

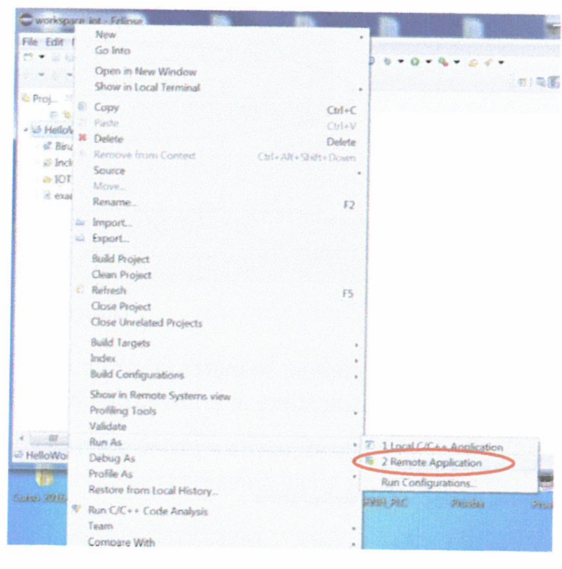


Figura 397

En la ventana que aparece, y desde la carpeta *Menú*, se escribe un nombre (aquí se ha colocado IOT2000) a la conexión, se marca el TCF Agent y se indica la ruta *(path)* donde se va a dejar el programa que funcionará en el SIMATIC IoT2040. Dicha ruta se acaba con el nombre del proyecto que, en este caso, es Primer_Programa. La ruta será */home/root/Primer_Programa*. El proceso se describe en la figura 398.

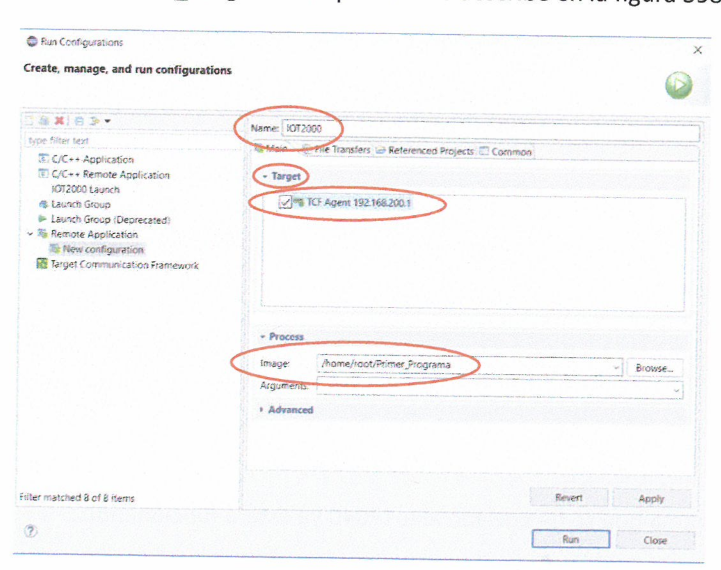


Figura 398

Seguidamente se accede a la pestaña *File Transfers,* donde se debe indicar la ruta del lugar del ordenador en el que se encuentra el proyecto, es decir, *workspace_iot,* y se vuelve a repetir la ruta del destino (del IoT2040). Se ha de acceder mediante *Add.* En la figura 399 se indica dicho proceso.

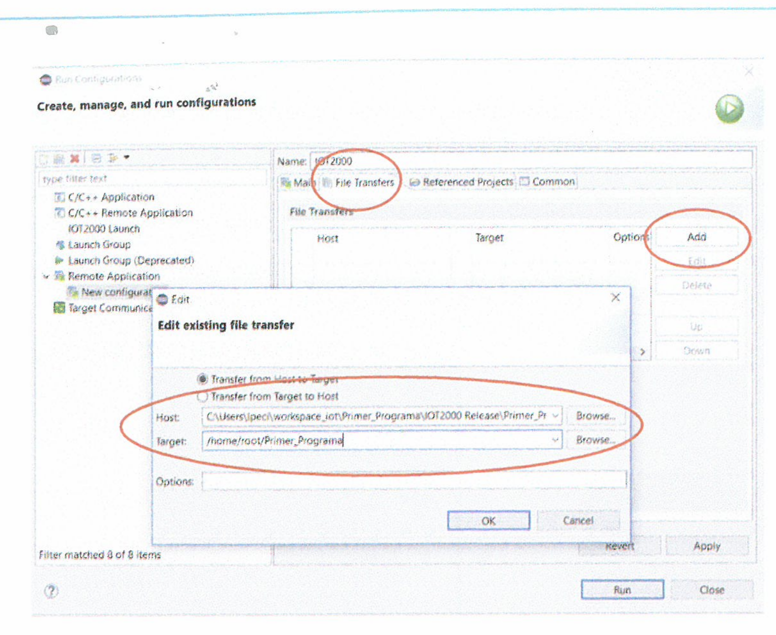


Figura 399

Ahora, desde la pestaña *Common,* se marca la casilla de *Run*, se pulsa sobre *Apply* para guardar las características de la conexión y se transfiere al IoT2040 mediante *Run* (figura 400).

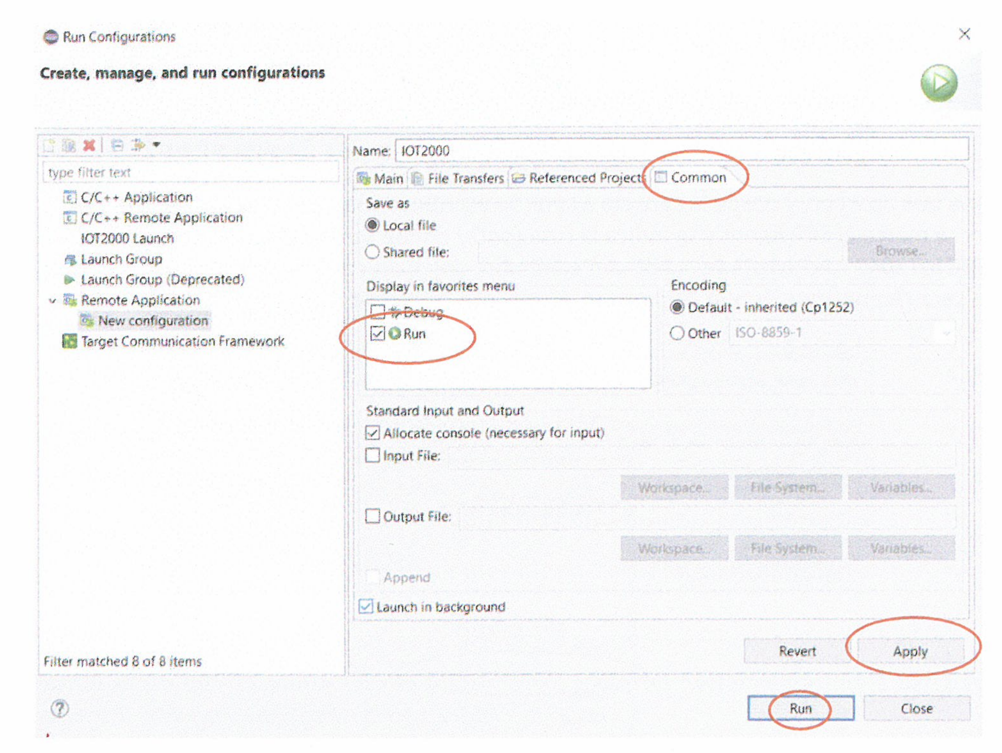


Figura 400

De esta forma, el programa que se ha enviado al IoT2040 ya estará funcionando.

Si todo ha ido bien, se verá parpadear el cursor desde la *Pestaña terminal*, como se comprueba en la figura 401.

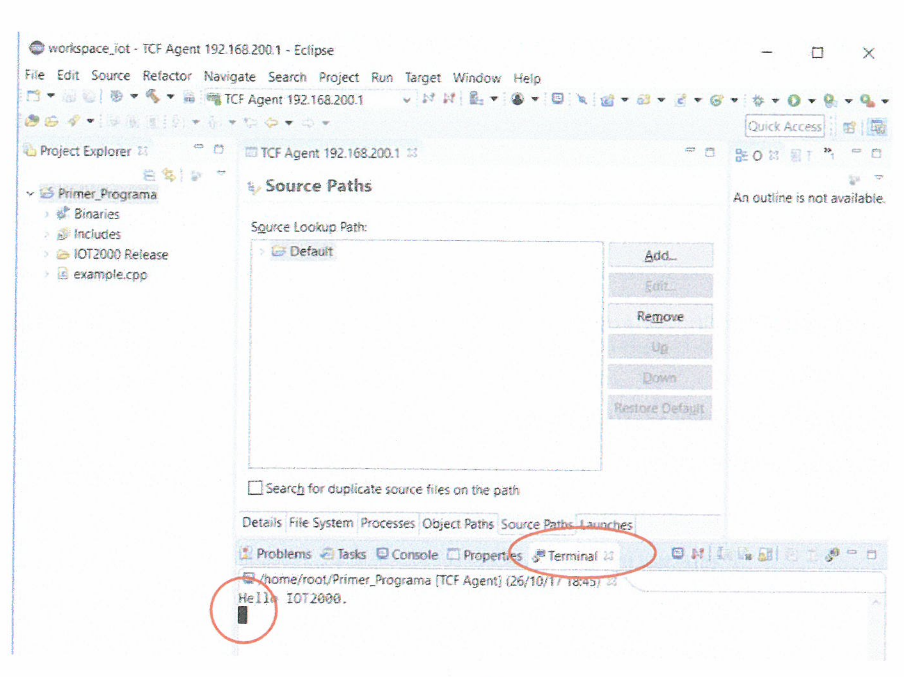


Figura 401

Si sale un mensaje de error como el de la figura 402 se debe volver a enviar desde el menú *Run As/Run Configurations...* y pulsar de nuevo sobre *Run.* En la figura 402 se representa dicha situación.

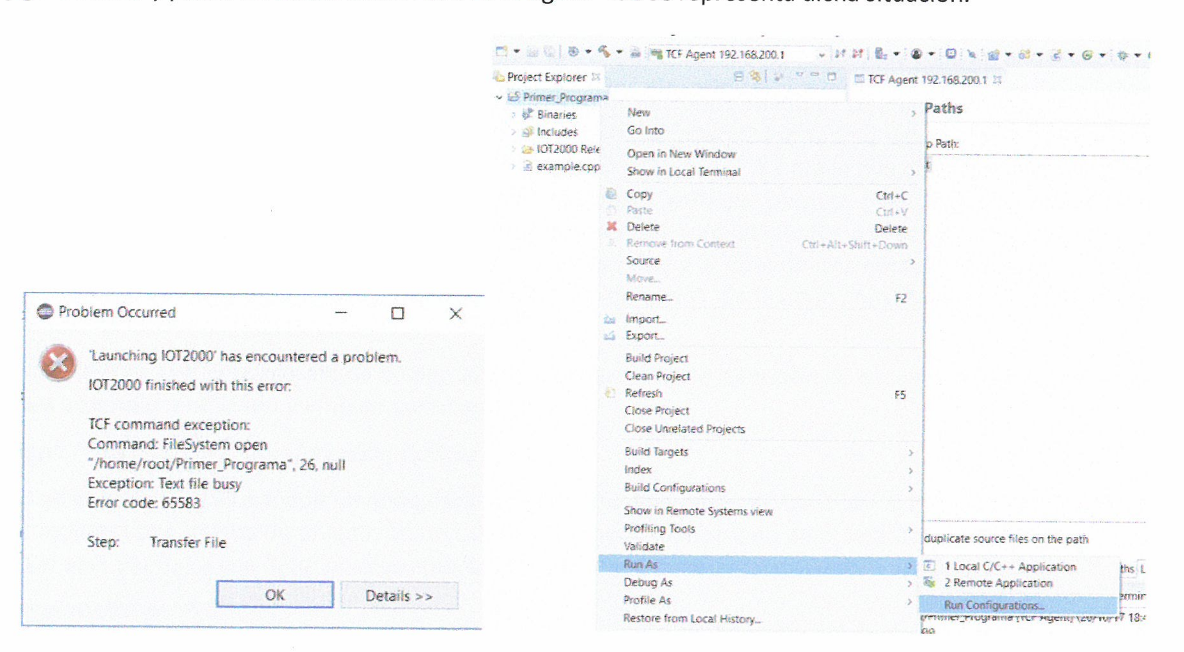


Figura 402

El programa que se ejecuta está escrito en C++ y es el siguiente:

```
#include <iostream>
using namespace std;
#include "mraa.hpp"
using namespace mraa;
int main(void) {
        cout << "Hello IOT2000." << endl;
        Gpio* d_pin = NULL;
        d_pin = new mraa::Gpio(13, true, false);
        d_pin->dir(mraa::DIR_OUT);
        for (;;) {
                d_pin->write(1);
                sleep(1);
                d_pin->write(0);
                sleep(1);
        }
        return 0;
}
```

Para comprobar este ejercicio se puede conectar un diodo led en serie con una resistencia de 330 ohmios entre masa y el pin 13 del pinout del Simatic IoT2040. También se puede utilizar el led USER del IoT2040, que funciona en paralelo con el pin 13.

27. IoT2040 Conexión con PLC. Lectura

Enunciado

En el ejercicio anterior se ha configurado el SIMATIC IoT2000, y programado, como dispositivo de programación de alto nivel. Ahora se va a conectar con un PLC S7-1500, se leerá un dato del PLC y se guardará en el dispositivo IoT2000.

Material necesario

1. Una pasarela SIMATIC IoT2040.
2. PLC S7 1516.
3. Una fuente de alimentación entre 9 y 32 V DC.
4. Un PC.
5. Un cable de Ethernet.
6. *Software* PuTTY.
7. Conexión a Internet.
8. *Switch* Scalance u otro.

Realización

Para la realización de este ejercicio, se va a utilizar la herramienta web Node-RED. Se trata de una plataforma web que convierte en realidad el hecho de interconectar dispositivos a Internet, es decir, que con ella se consigue implementar aplicaciones del Internet de las cosas. Ha sido creada por IBM (por el departamento de tecnologías emergentes). Node-RED conecta tanto dispositivos entre sí como entre servicios *online*.

Node-RED dispone de una interfaz basada en un navegador web que permite crear **flujos** de eventos e interconectarlos todos ellos a través de una plataforma sencilla y amigable. Ha sido construido en **node.js**, lo que le faculta funcionar justo al borde de la red o en la nube y le dota de una notable flexibidad. El gestor de paquetes de node (npm) hace posible la ampliación de nodos. Cada nodo constituye un dispositivo o un servicio que pueden interconectarse.

Node-RED es una herramienta de código abierto. En la siguiente dirección se encuentra todo lo necesario para aprender a utilizarlo y ampliar conocimientos:

https://nodered.org/docs/creating-nodes/.

En este ejercicio se va a añadir un nodo, facilitado por Siemens, para interconectar el PLC al SIMATIC IoT2000. Se realizará una lectura de un dato en un DB del PLC y será almacenado en el SIMATIC IoT2040. El siguiente paso será dar uso a ese dato, por ejemplo, colgándolo en la nube.

Preparación del PLC

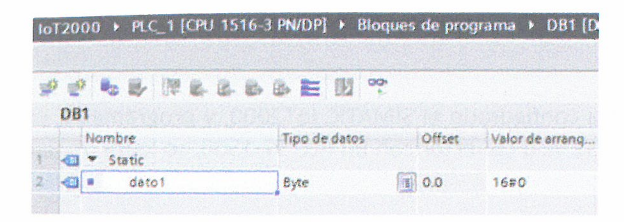
Para comprobar que los datos son leídos correctamente por el SIMATIC IoT2040, se elabora un sencillo programa que incremente en una unidad el contenido de un dato de un DB, cada vez que se activa un pulsador. Mediante otro pulsador, se pondrá a cero dicha variable del DB.

El programa del OB1 es el siguiente:

```
        U     "Tag_1"            %E0.0
        FP    "Tag_4"            %M31.0
        SPB   11

        U     "Tag_3"            %E0.1
        SPB   12
        BEA
  11:   L     1                  1
        L     "DB1".dato1        %DB1.DBB0
        +I
        T     "DB1".dato1        %DB1.DBB0
        BEA
  12:   L     0                  0
        T     "DB1".dato1        %DB1.DBB0
```

El DB1 es:



Además, se debe activar el permiso de acceso vía comunicación PUT/GET. Esto se recuerda en la figura 404.

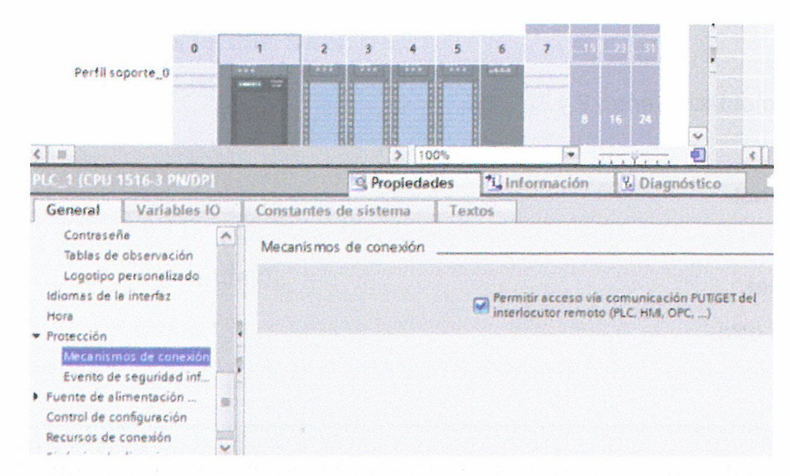


Figura 404

También es necesario utilizar el DB de manera que se pueda acceder de forma no optimizada, por lo que habrá que **DESMARCAR** dicha opción, que, por defecto, se encuentra marcada como se aprecia en la figura 405. Para acceder a la pantalla, se pulsa con el botón derecho en el nombre del DB1, desde el árbol del proyecto.

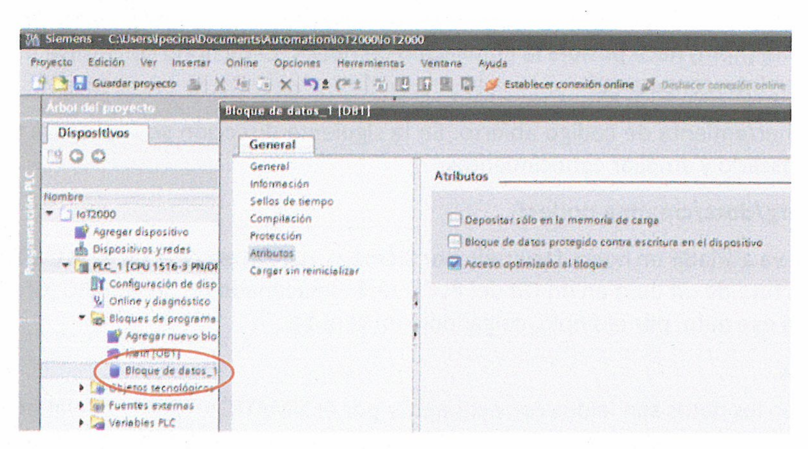


Figura 405

Instalación y Arranque del NODE-RED del PLC S7

Lo primero que se ha de hacer en el dispositivo SIMATIC IoT2040 es instalar el nodo S7, creado por Siemens. Este Node-RED se halla preinstalado en la imagen que se cargó en la tarjeta SD en el ejercicio anterior.

Resulta muy importante que el IoT2000 disponga de una conexión a Internet. En el caso de que no la tuviera, no se podría realizar correctamente la instalación. El IoT2000 debe contar con una conexión directa de Internet; es decir, en el caso del IoT2040, el cable de Ethernet, con la conexión de Internet, debe comunicarse con la entrada X2. En la figura 406 se representa dicha situación. El cable negro conecta con el PC y el PLC mediante un *Switch*.



Figura 406

En el caso de disponer de un IoT2020, al no contar con dos conexiones RJ45, debe utilizarse un dispositivo wifi/USB.

Ahora se abre el *software* PuTTy con las características de conexión vistas anteriormente para el IoT2000. Una vez dentro, tecleando la clave *root*, se accede al directorio *node_modules;* para ello, se escribe la siguiente ruta: cd /usr/lib/node_modules (¡cuidado!: entre **cd** y **/** hay un espacio).

Una vez dentro del directorio, se teclea el comando «npm» para instalar el nodo: npm install node-red-contrib-s7.

En la figura 407 se puede comprobar el resultado al instalar el nodo S7. Se debe esperar un tiempo hasta que salga la pantalla.

```
root@iot2000:/# npm install node-red-contrib-s7
/
`-- node-red-contrib-s7@1.4.0
  `-- nodes7@0.2.5

npm WARN enoent ENOENT: no such file or directory, open '/package.json'
npm WARN !invalid#1 No description
npm WARN !invalid#1 No repository field.
npm WARN !invalid#1 No README data
npm WARN !invalid#1 No license field.
root@iot2000:/#
```

Figura 407

Si no se tuviera acceso a Internet, saldría la pantalla de error (figura 408):

```
192.168.200.1 - PuTTY
19 Aug 10:08:40 - [info] Server now running at http://127.0.0.1:1880/
19 Aug 10:08:40 - [info] Starting flows
19 Aug 10:08:40 - [info] Started flows
install node-red-contrib-s7
install: missing destination file operand after 'node-red-contrib-s7'
Try 'install --help' for more information.
root@iot2000:/# cd /usr/lib/node_modules
root@iot2000:/usr/lib/node_modules# npm install node-red-contrib-s7
npm ERR! Linux 4.4.18-yocto-standard
npm ERR! argv "/usr/bin/node" "/usr/bin/npm" "install" "node-red-contrib-s7"
npm ERR! node v6.9.2
npm ERR! npm  v3.10.9
npm ERR! code EAI_AGAIN
npm ERR! errno EAI_AGAIN
npm ERR! syscall getaddrinfo

npm ERR! getaddrinfo EAI_AGAIN registry.npmjs.org:443
npm ERR!
npm ERR! If you need help, you may report this error at:
npm ERR!     <https://github.com/npm/npm/issues>

npm ERR! Please include the following file with any support request:
npm ERR!     /usr/lib/node_modules/npm-debug.log
root@iot2000:/usr/lib/node_modules#
```

Figura 408

El siguiente paso consiste en arrancar el nodo S7 creado. Para ello, y desde el directorio raíz (se teclea cd / para ir al directorio raíz), se escribe: node /usr/lib/node_modules/node-red/red & (entre node y / hay un espacio). Transcurrido un tiempo, aparecerá la pantalla de la figura 409.

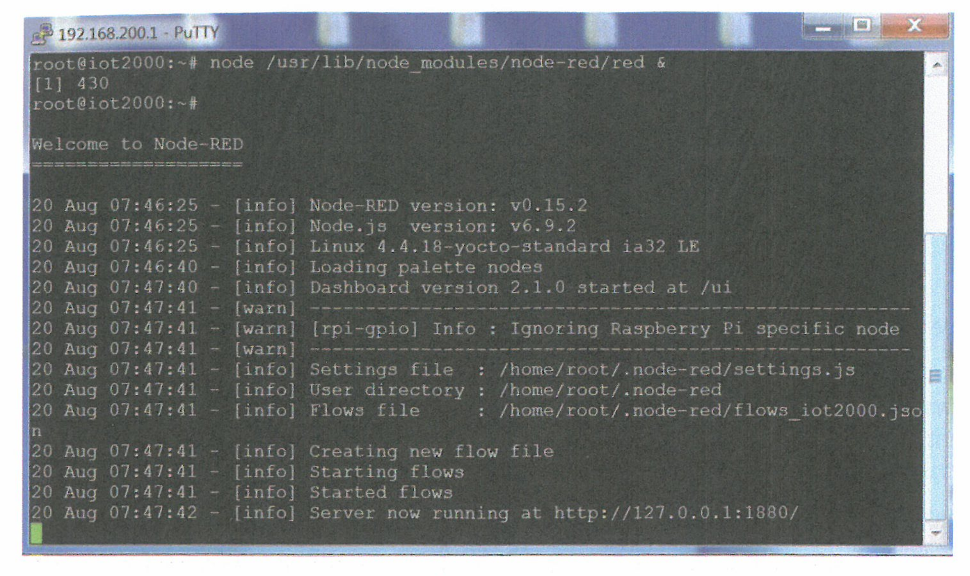


Figura 409

Apertura de la interfaz Node-RED en un explorador de Internet

Ahora es el momento de abrir la interfaz de Node-RED en la web. Para ello, se teclea en un explorador la dirección del IoT2040 seguido del puerto 1880, es decir, 192.168.200.1:1880. La pantalla que aparece se indica en la figura 410.

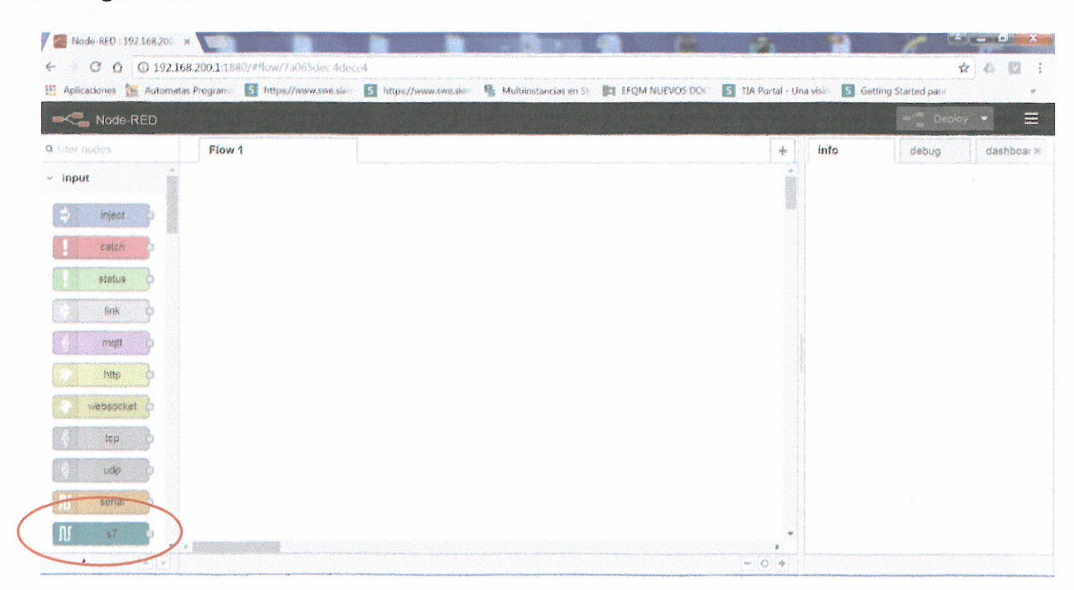


Figura 410

En la interfaz de Node-RED, en su parte izquierda, se distinguen los nodos existentes. Entre ellos aparece el nodo creado S7. El siguiente paso es arrastrar hasta la parte central de la interfaz el nodo S7. Seguidamente, se pulsa dos veces sobre el nodo para que aparezca, a la derecha, la ventana de edición. En la figura 411 aparece el nodo S7 en el área central, así como la ventana de edición.

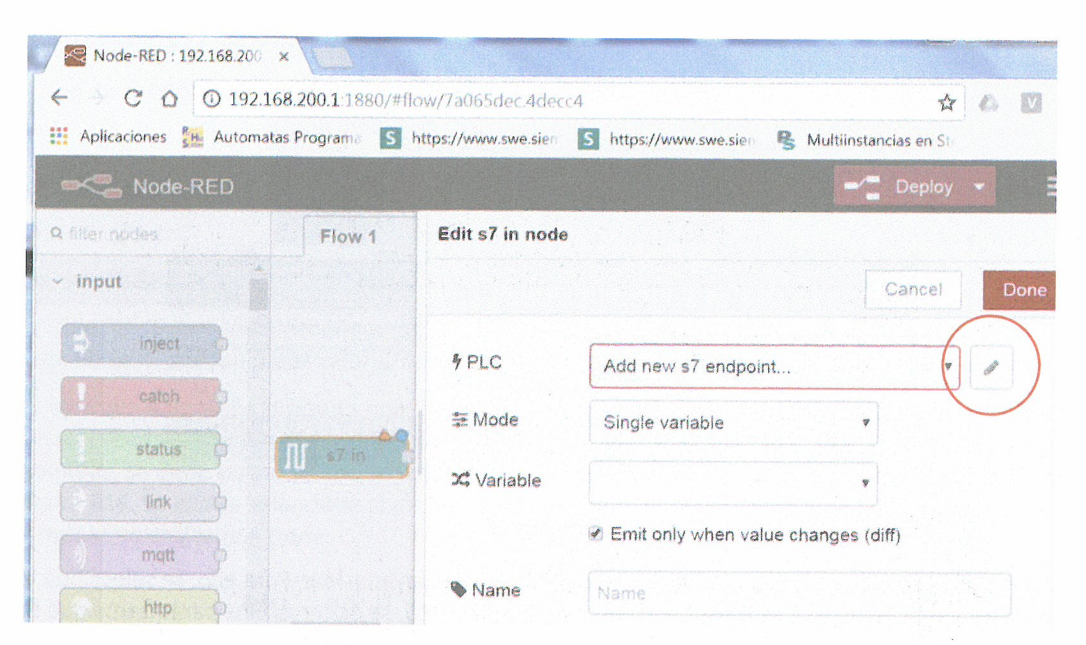


Figura 411

En primer lugar, se indica la dirección del PLC con el que se va a conectar y algún parámetro más. Para ello, se selecciona el icono lapicero. Entonces aparecerá la ventana de la figura 412, donde se coloca la IP y el número de *rack* y *slot* que ocupa el PLC en el proyecto de TIA PORTAL.

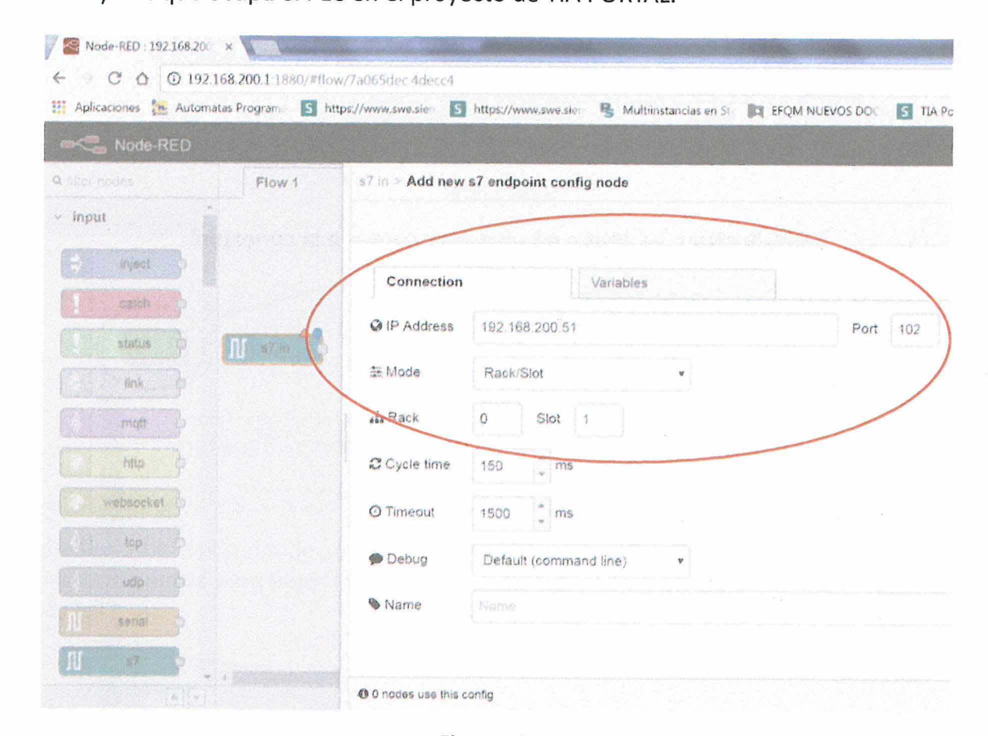


Figura 412

Para conocer el número de *rack* y *slot* del PLC, se debe ir a las propiedades del PLC. Se aprecia dicha situación en la figura 413.

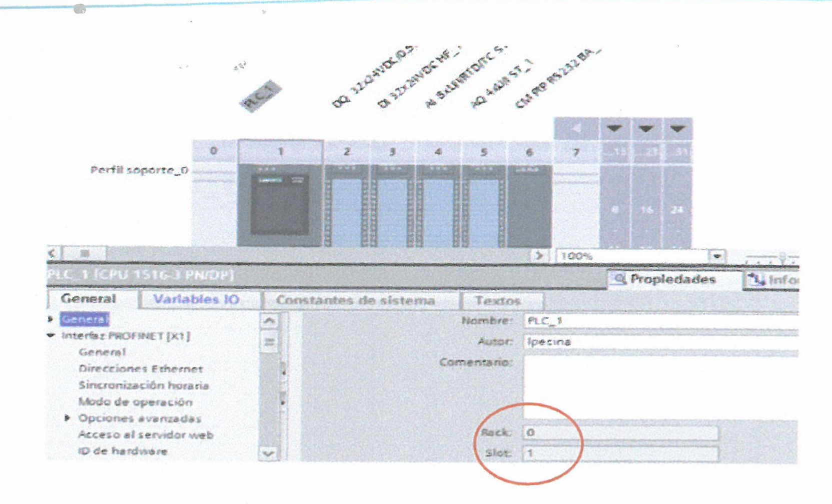


Figura 413

El siguiente paso consiste en indicar las variables del PLC que se vayan a leer. Para ello, se selecciona la pestaña *Variables*. En la figura 414 se muestra el DB y, en la figura 415, se observan los parámetros que han de completarse.

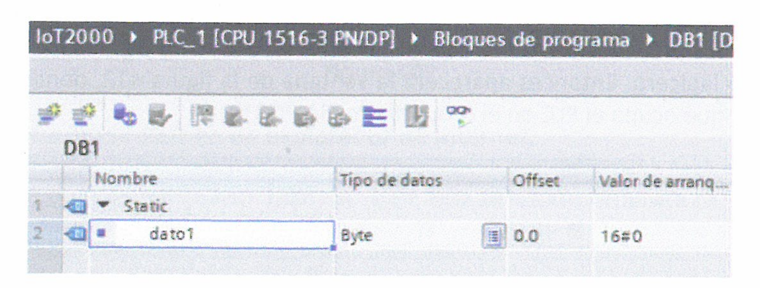
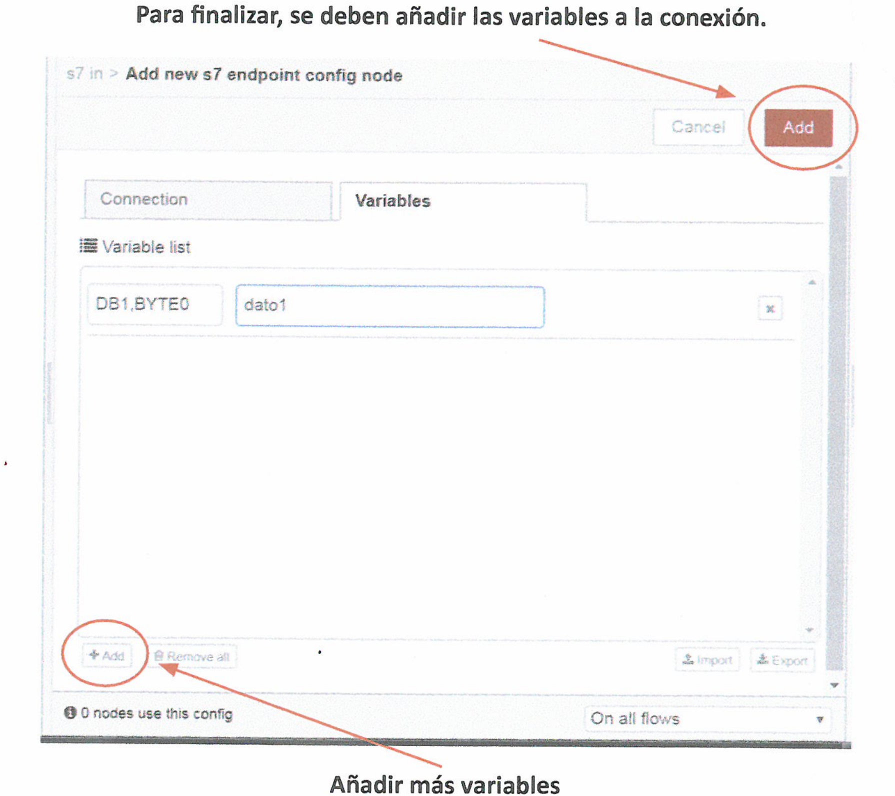


Figura 414

Para finalizar, se deben añadir las variables a la conexión.



Añadir más variables

Figura 415

Las variables del DB se ponen en virtud del siguiente formato:

N.°DB,Tipo de datoDirección Offset de la variable en el DB; en este ejemplo, DB1,BYTE0.

N.° de DB: DB1.

Tipo de dato: *byte*.

Dirección de *offset*: 0.

Además, se añadirá el nombre de la variable, tal como aparece en el DB.

Si la variable fuese de tipo bit (Bool), su codificación sería: DB1,X1.0 (siendo la dirección de *offset* la 1.0). En este caso, no se le pueden pasar valores numéricos, es decir, unos o ceros, si no *true* o *false*.

En este ejemplo solo se ha utilizado una variable, pero, si se desea añadir la lectura de más variables, se puede hacer mediante la opción *Add*, ubicada en la parte inferior izquierda. Para añadir todas las variables seleccionadas, se debe pulsar la opción *Add* del área superior derecha. A continuación, surgirá la ventana de la figura 416. En ella se selecciona el modo de visualización de las variables y se le da al nodo (opcional). Para terminar, se selecciona *Done*.

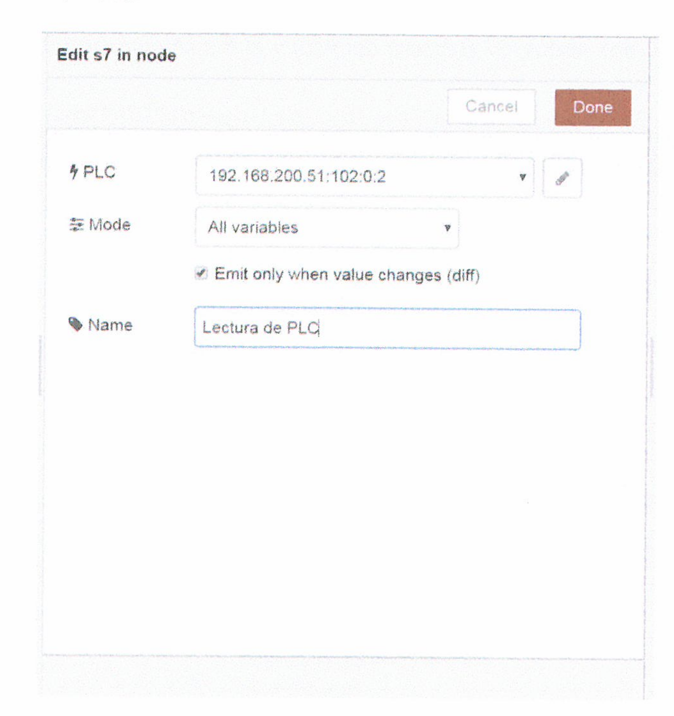


Figura 416

Ahora se debe buscar el nodo donde se va a almacenar el valor leído desde el PLC. Se deberá realizar *scroll* en el menú de la izquierda para bajar hasta el menú *storage*. Allí se arrastra el nodo *file*, como se observa en la figura 417.

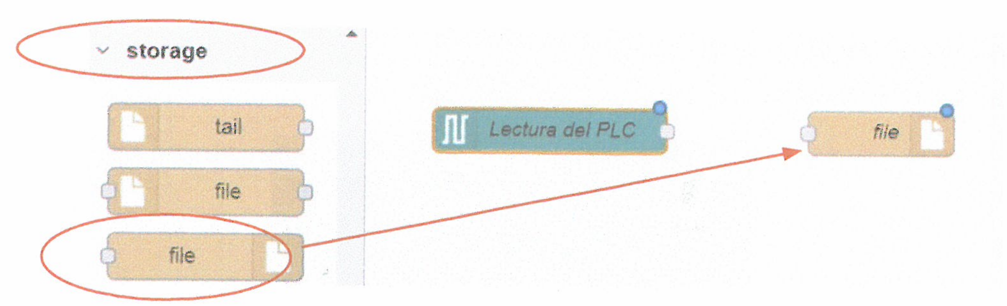


Figura 417

A continuación, con doble pulsación sobre el nodo *file*, se accede a la edición de este, como se puede comprobar en la figura 418.

Se introducen los datos, tal como aparecen en la figura 418 y, después, se selecciona *Done*.

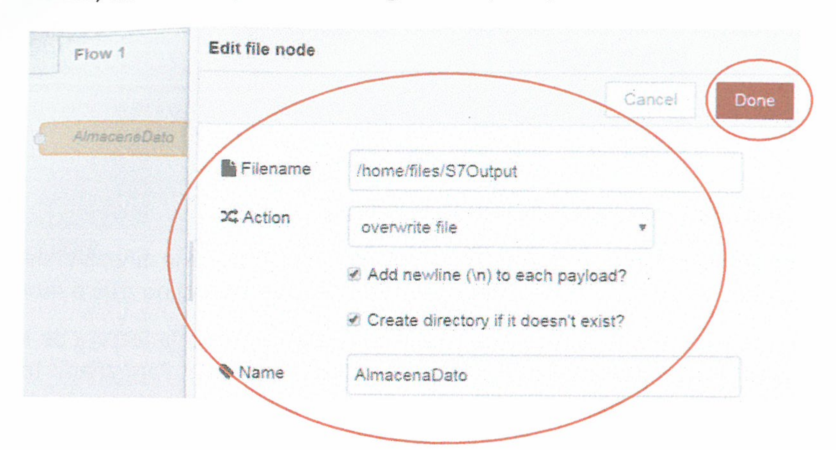


Figura 418

Seguidamente se unen los dos nodos, S7 *(Lectura del PLC)* y el *file (AlmacenaDato)*, como se indica en la figura 419.

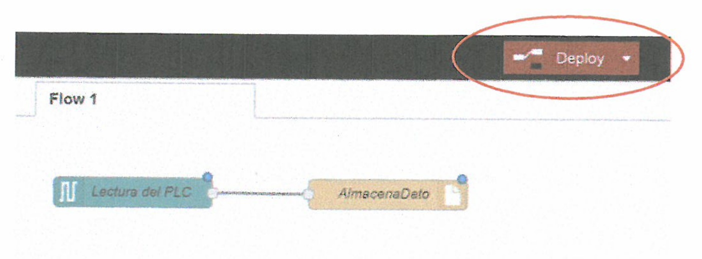


Figura 419

Para que arranque, se acciona *Deploy*.

Ahora se observarán en PuTTy los valores de la variable del PLC, si se realizan los pasos expuestos a continuación.

Primero, se ha de acceder al directorio *files*:

cd /home/files (espacio entre cd y /).

Allí se escribe el comando

«ls» (ele y ese).

Aparecerá en la pantalla «S7Output»

Luego se teclea:

«catS7Output»

Ya se podrá leer el valor de la variable deseada.

En la figura 420 se indica dicha situación.

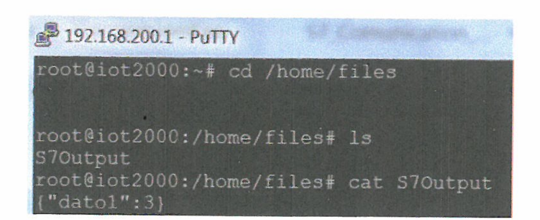


Figura 420

Cada vez que se desee leer el valor de la variable del DB, se tecleará «catS7Output».

28. IoT2040 Conexión con PLC. Escritura

Enunciado

En el ejercicio anterior se ha realizado la lectura de un dato del PLC. Ahora se va a conectar con un PLC S7-1500 y se escribirá un dato del PLC desde el IoT2040.

Material necesario

1. Una pasarela SIMATIC IoT2040.
2. PLC S7 1516.
3. Una fuente de alimentación entre 9 y 32 V DC.
4. Un PC.
5. Un cable de Ethernet.
6. *Software* PuTTY.
7. Conexión a Internet.
8. *Switch* Scalance u otro.

Realización

Este ejercicio constituye una continuación del anterior. Se le va a añadir la escritura en un dato de tipo *byte* del DB1 ya creado en ese ejercicio. El dato no debe utilizarse en el OB1.

Se ha de tener mucho cuidado con la activación desde el exterior de variables del PLC, si este está en pleno funcionamiento.

El punto de partida para este ejercicio será arrancar el programa PuTTy y, una vez allí, el Node-RED. Para ello, se vuelve a escribir «node /usr/lib/node_modules/node-red/red &» (entre node y / hay un espacio). Una vez hecho esto, se abrirá Node-RED en el navegador colocando la dirección y puerto siguiente: 192.168.200.1:1880.

Se puede comprobar el nodo de salida llamado S7, como se observa en la figura 421.

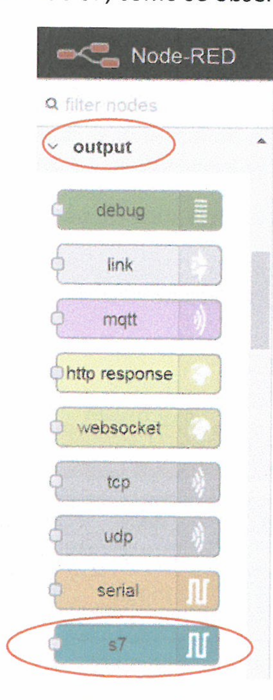
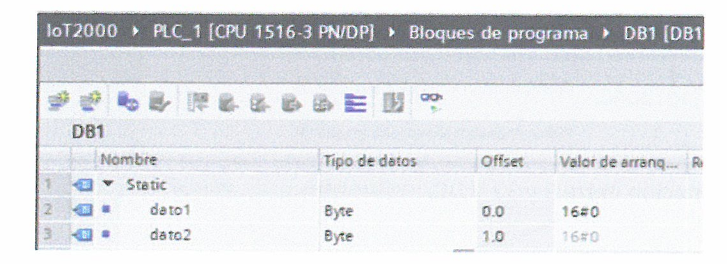


Figura 421

Node-RED funciona pasando de un bloque a otro (nodo) una información denominada **msg.payload**. Así va fluyendo entre los diferentes nodos; algunos nodos transformarán la información. Uno de esos nodos es el de *Función*. En la función se puede escribir un programa en C que maneje ese dato que fluye, con lo que se modifica su valor. De esa función saldrá otro **msg.payload**.

En este ejercicio se va a añadir una nueva variable en el DB1. Será de tipo *byte*. En la figura 422 se muestra el DB1.

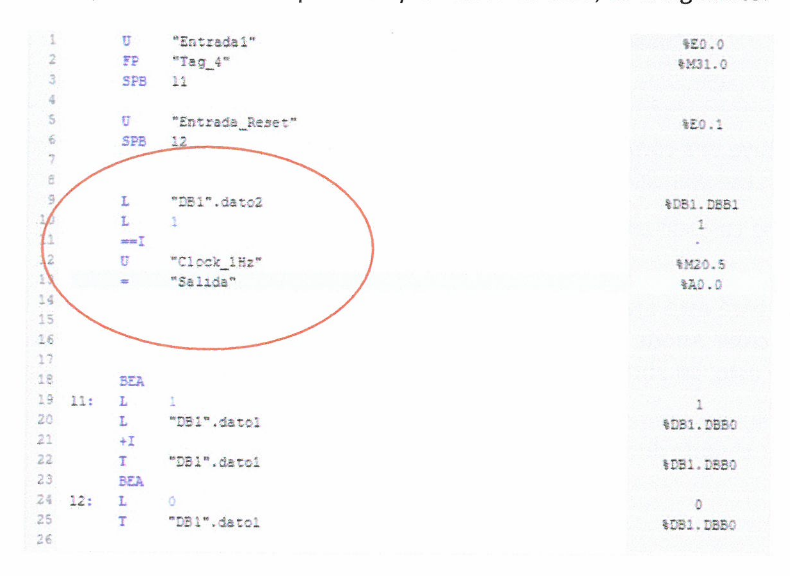

IoT2000 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Bloques de programa ▸ DB1 [DB1

DB1

	Nombre	Tipo de datos	Offset	Valor de arranq...
1	▼ Static			
2	▪ dato1	Byte	0.0	16#0
3	▪ dato2	Byte	1.0	16#0

Figura 422

En el OB1 se añade igualmente código nuevo. Ahora el programa lo que encenderá una luz intermitente cuando este nuevo *dato2* se halle a 1. Este dato lo pondrá a 1 el IoT2040 cuando el *dato1* sea igual o mayor a 5. El programa del OB1, añadiendo la comparación y la marca de ciclo, es el siguiente:



```
      U     "Entrada1"                %E0.0
      FP    "Tag_4"                   %M31.0
      SPB   11

      U     "Entrada_Reset"           %E0.1
      SPB   12


      L     "DB1".dato2               %DB1.DBB1
      L     1                         1
      ==I
      U     "Clock_1Hz"               %M20.5
      =     "Salida"                  %A0.0




      BEA
11:   L     1                         1
      L     "DB1".dato1               %DB1.DBB0
      +I
      T     "DB1".dato1               %DB1.DBB0
      BEA
12:   L     0                         0
      T     "DB1".dato1               %DB1.DBB0
```

En el Node-RED, se ha de incluir una función de forma que, cuando el *dato1*, leído desde el PLC, sea igual o mayor que 5, pase al **msg.payload** un 1. Si es menor de 5, pasará al **msg.payload** un 0.

La función se encuentra en el lado izquierdo, en la zona de nodos, como puede verse en la figura 423.

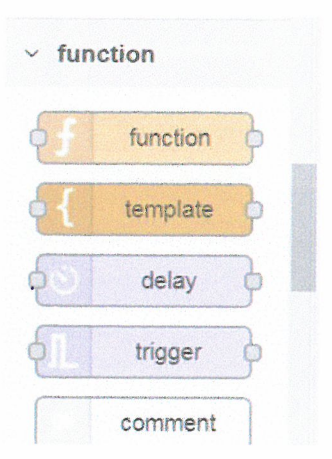


Figura 423

Se arrastra al *clipboard* (parte central de flujo de nodos) y se escribe el programa tocando dos veces sobre el nodo *function*. El programa que se debe escribir es el que aparece en la figura 424.




```
if (msg.payload >= 5)
msg.payload = 1;
else
msg.payload = 0;
return msg;
```

Figura 424

Ahora se añade el nodo de salida de S7, que se ha cargado en el ejercicio anterior al instalar el nodo S7. Se arrastra el nodo S7 desde los nodos de salida *(Output).* Pulsando dos veces sobre él, se adicionan los parámetros de la conexión y la nueva variable. Dicha variable constituye el nuevo dato creado en el DB1; se llama, en este caso, *dato2*. Se seguirá el mismo criterio de definición de la variable que en el ejercicio anterior. Esto se muestra en la figura 425.

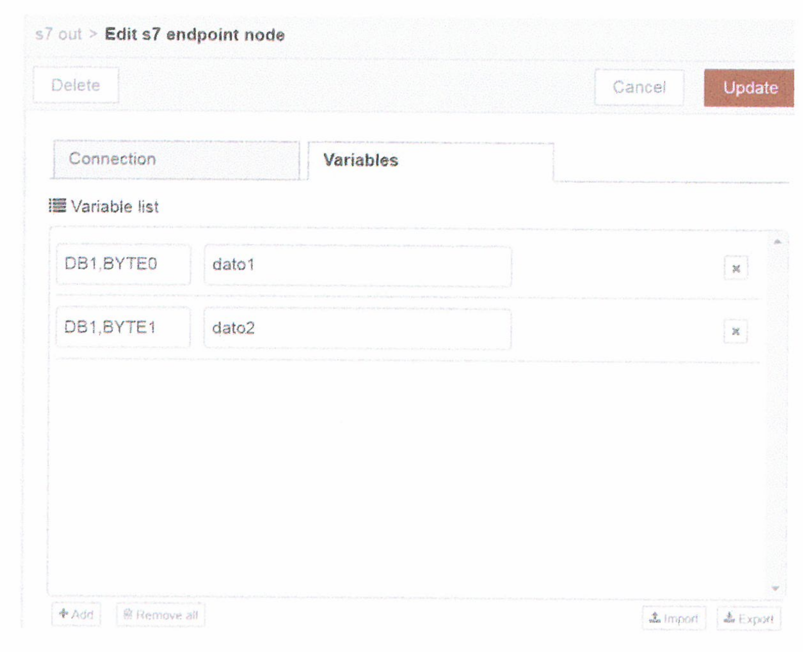


Figura 425

Lo mismo debe hacerse con la conexión (figura 426).

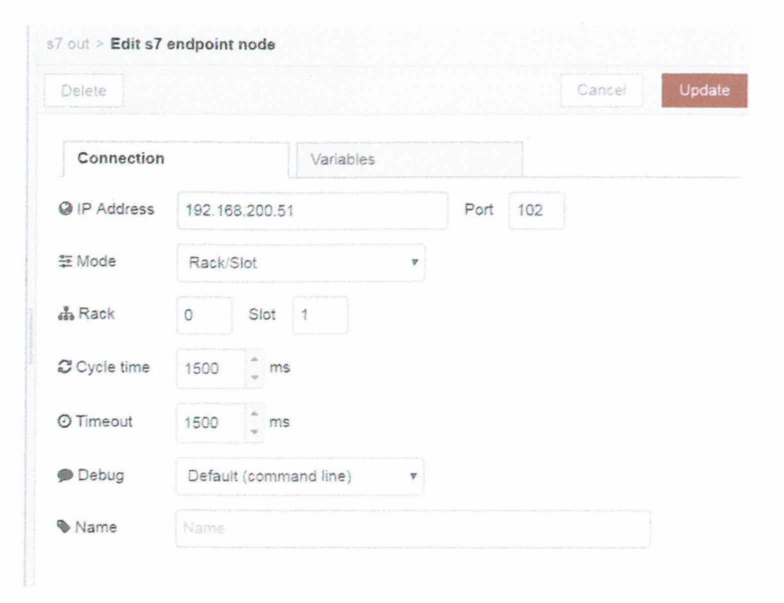


Figura 426

El resultado del *Flow* final se aprecia en la figura 427. Para que arranque, se acciona *Deploy*.

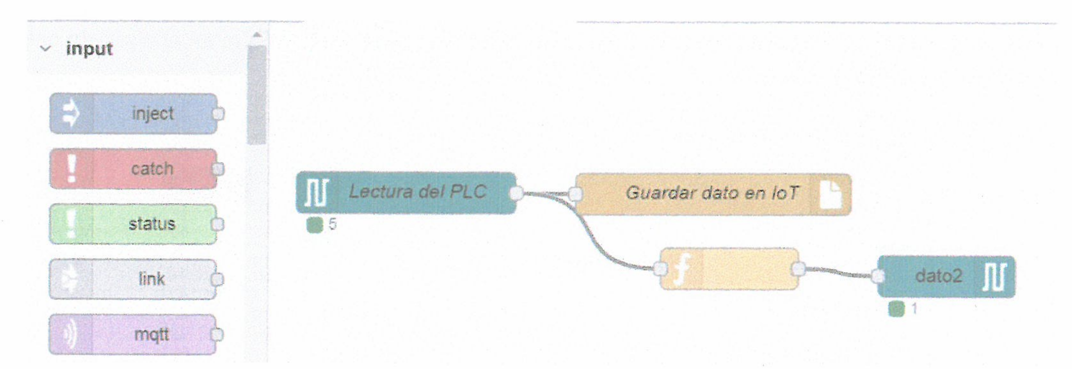


Figura 427

29. IoT2040 *e-mails* y Twitter

Enunciado

En este ejercicio se van a utilizar los datos que se han adqurido desde el IoT2040 y se van a enviar por *e-mail,* así como también a una cuenta de Twitter.

Material necesario

1. Una pasarela SIMATIC IoT2040.
2. PLC S7 1516.
3. Una fuente de alimentación entre 9 y 32 V DC.
4. Un PC.
5. Un cable de Ethernet.
6. *Software* PuTTY.
7. Conexión a Internet.
8. *Switch* Scalance u otro.

Realización

Se ha comprobado que el uso de Node-RED abre una infinidad de posibilidades. Ahora se va a enviar una información del PLC a una dirección de correo electrónico y a una cuenta de Twitter.

En primer lugar, se introduce el IoT2040 con la fecha y hora actual, si no lo estuviera ya. Para ello, se teclea la orden «date» en el *software* PuTTy. Aparecerán así la hora y la fecha; si no fuesen correctas, habría que corregirlas. Para ello, se pone «date *MMDDhhmmYYYY*»; por ejemplo, para escribir 15 de noviembre de 2017 a las 12 h 30 min, se introduciría «date *111512302017*». Si este punto no se pusiera correctamente, no se podrían enviar *e-mails* ni datos a la cuenta de Twitter.

En este ejercicio se va a realizar el envío, por correo electrónico y Twitter, de un texto solo cuando el contenido de la variable *dato1* del DB1, del ejercicio 27, sea igual o mayor de 5. En este caso, como se ha hecho en el ejercicio anterior, también lucirá una salida de forma intermitente. Se visualizará igualmente en el monitor de la interfaz de Node-RED *(Debug)* determinada información.

El texto que va a aparecer en el *e-mail* y en Twitter es *La variable dato1 ha llegado a 5;* también se incluirá fecha y hora.

Los nodos de los dos ejercicios anteriores permanecen como están. Ahora se van a añadir los nodos necesarios para que se envíe un *e-mail* y un tuit, y solo uno, cuando la variable *dato1* llegue a 5 o, lo que es lo mismo, que la variable *dato2* sea 1.

Primero se realiza la parte correspondiente al envío del *e-mail*. Para ello, se coloca un nodo función, donde se pondrá *msg.payload* a *true,* cuando el valor de la variable *dato1* sea 5; si no fuese igual a 5, pasaría a *msg. payload* el valor que tenga la variale *dato1*. La función será:

```
if (msg.payload == 5) msg.payload = true;
else msg.payload= msg.payload;
return msg;
```

Como se ha dicho, el *payload* es el dato que pasa entre los bloques. Ahora, en función de si el *msg.payload* es *true* o cualquier otro valor numérico, hará cosas diferentes, para lo que se deben abrir dos caminos, lo cual se realizará con la función *Switch*. En la figura 428 se puede comprobar el lugar donde se encuentra y su configuración. Como se observa, si el *msg.payload* es *true,* seguirá el camino 1 y, si es menor de 5, seguirá el camino 2.

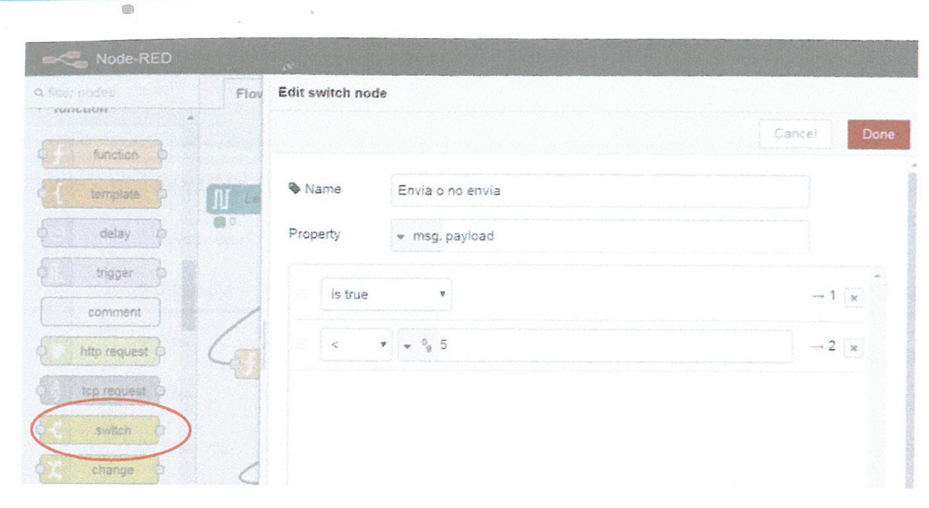


Figura 428

El camino uno conecta con otra función que prepara el mensaje y el objeto del correo. Se puede observar en la figura 429. En este caso, se trata de un texto al que se le añade la fecha y la hora:

Date( ).toString( ).

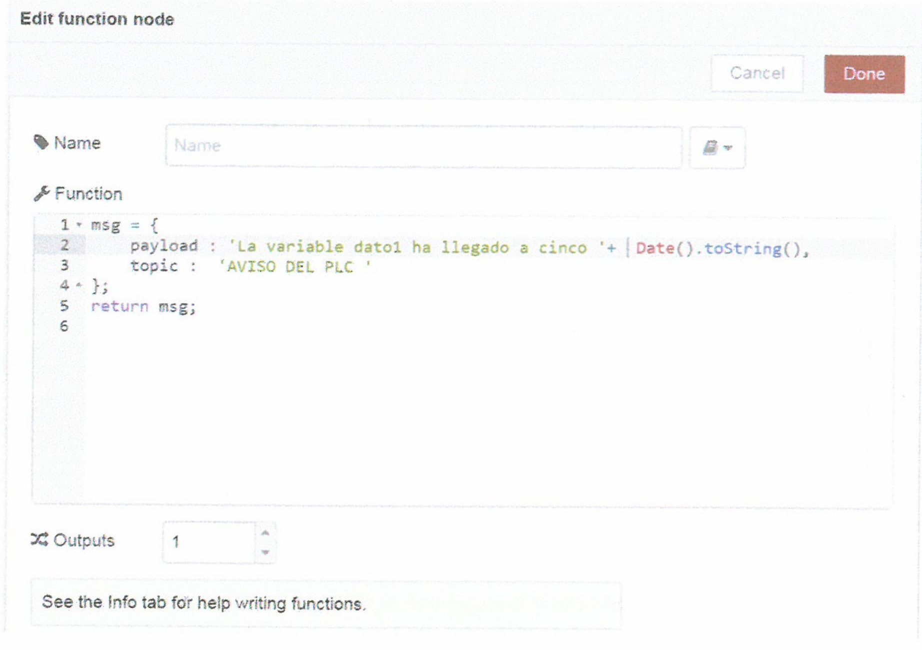


Figura 429

Ahora se coloca el nodo de enviar correos. Solo se pueden mandar con el servidor de Gmail. Este nodo se encuentra en la sección de Social, en el lado izquierdo. En la figura 430 se aprecia dónde se encuentra y la configuración de dicho nodo. Se debe disponer de una cuenta de Gmail para poder utilizar el nodo. Se debe introducir el servidor de Gmail (smtp.gmail.com), el puerto 465, el usuario de la cuenta Gmail y la contraseña. A esto se añade la dirección de correo de la persona a quien va dirigido y, de forma opcional, se le da un nombre al nodo.

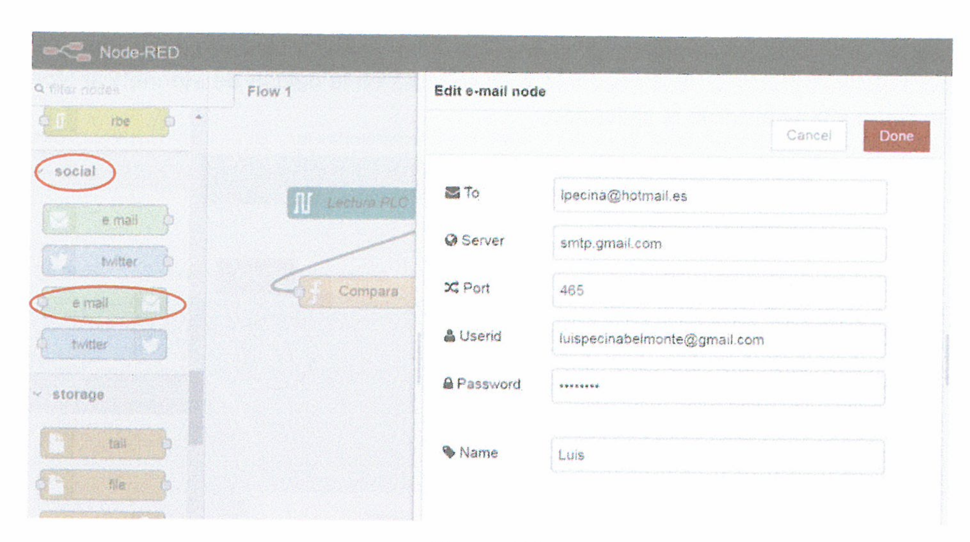


Figura 430

El segundo camino de la función *Switch* conecta con otra función que visualiza cierta información en el *Debug* de la interfaz de Node-RED. Se puede apreciar en la figura 431 cómo, si el *payload* es *true*, enviará al *Debug* (siguiente nodo) el texto: *La variable dato1 es mayor o igual que 5 pero ya se ha enviado email*. Con cualquier otro valor, se visualiza el contenido de la variable *dato1*.

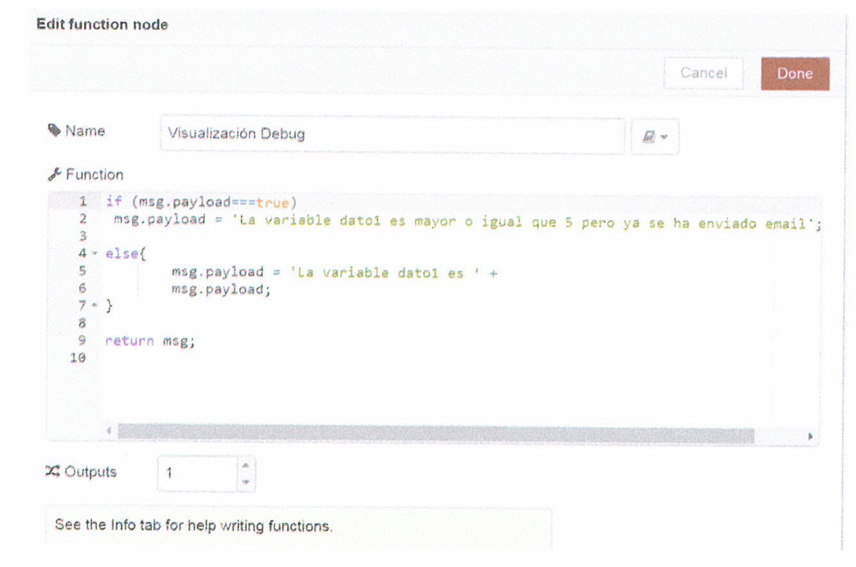


Figura 431

Ahora se conecta el nodo *Debug* que se encuentra en la sección de salidas, como se puede observar en la figura 432.

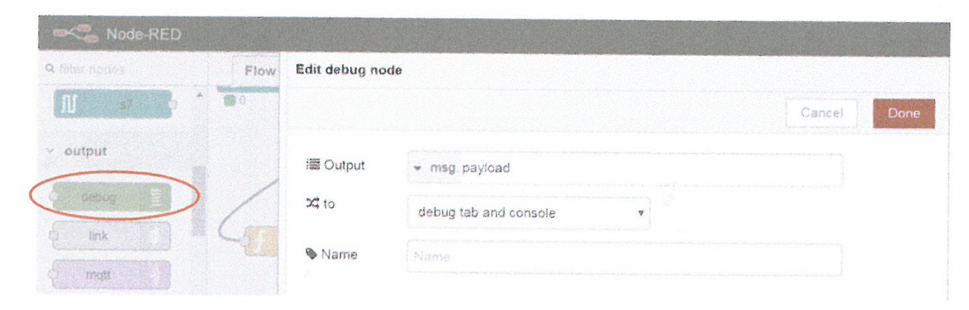


Figura 432

Para añadir el envío de la información a una cuenta de Twitter, se debe seleccionar el nodo de escribir en Twitter. Se coloca en paralelo al envío de *e-mails*.

Una vez abierto el nodo, se debe autenticar la cuenta. Para ello, según la figura 433, se pulsa sobre el lapicero. En la figura 434 se aprecia la nueva ventana y allí se pulsa en la opción de autentificar Node-RED en Twitter. Y, en la figura 435 se autoriza la aplicación.

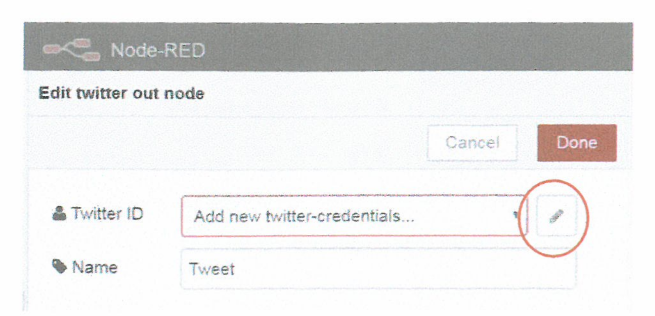


Figura 433

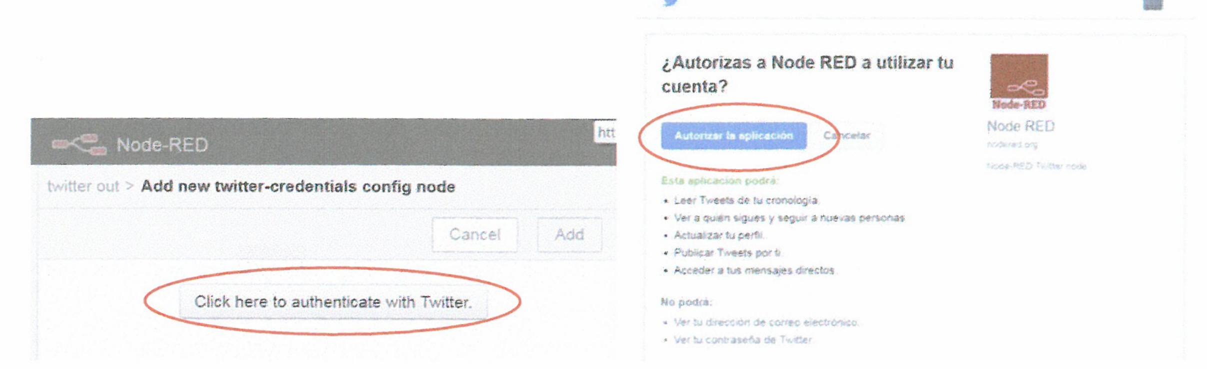


Figura 434

Figura 435

Ahora se debe añadir el nombre de la cuenta (figura 436) y un nombre, si se desea (figura 437).

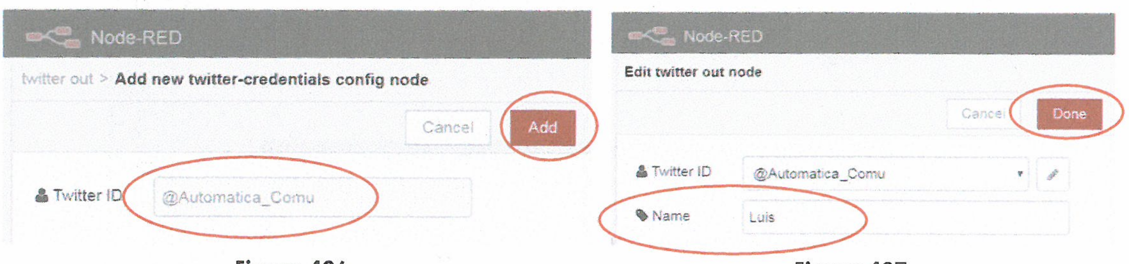


Figura 436

Figura 437

El *Flow* completo queda de la forma que se muestra en la figura 438.

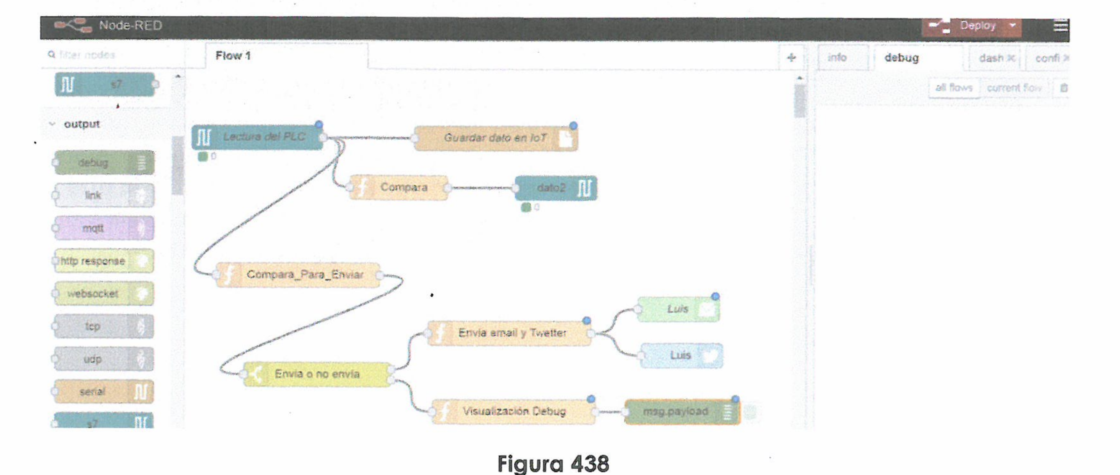


Figura 438

Ahora se arranca el *Flow* creado (pulsando en *Deploy*) y, cuando la variable del DB1, *dato1*, ha llegado a 5, se enviará un *e-mail* y un tuit. En las figuras 439 y 440 puede verse la recepción de dicho *e-mail* y tuit.



Figura 439

Figura 440

Si en alguno de los procesos de enviar *e-mails* o de autentificar la cuenta de Twitter sale algún error, se debe a no tener sincronizado el reloj del SIMATIC IoT2040.

Por último, si se desea monitorizar los nodos en la propia interfaz web de Node-RED, hay que activar la opción *Show node status* (figura 441).

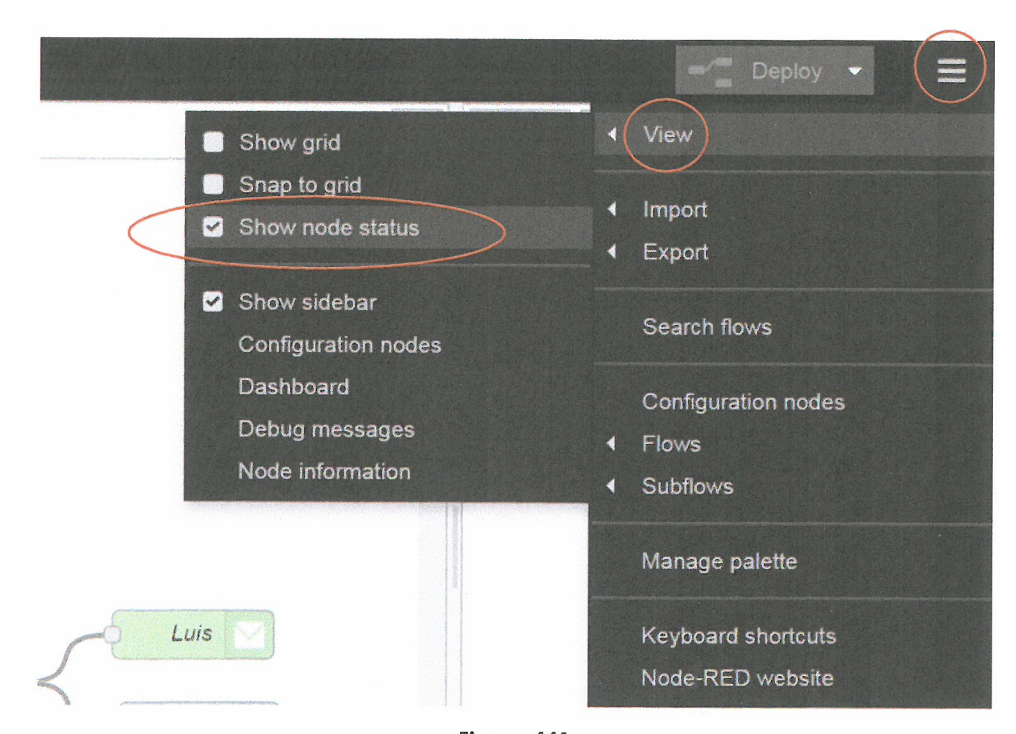


Figura 441

